Sistemas Informáticos
Curso 2003 / 04

Modelo de seguridad para una arquitectura de Servicios Web XML

Realizado por:

Náyade Bermúdez Brito
Abel Miguel Ibáñez Mojarro
Javier González del Castillo

Dirigido por:

Prof. José Jaime Ruz Ortiz
Dpto. Arquitectura de Computadores y Automática

Facultad de Informática
Universidad Complutense de Madrid

Índice

1. Autorización a la UCM.

Los autores de este proyecto, Náyade Bermúdez Brito, Javier González del Castillo y Abel Miguel Ibáñez Mojarro, autorizamos a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos (no comerciales) tanto la memoria como el código y el prototipo desarrollado.

Fdo. Náyade Bermúdez Brito

Fdo. Abel Miguel Ibáñez Mojarro

Fdo. Javier González del Castillo

2. Resumen.

El proyecto es un Servicio Web en .NET que optimiza una red de transporte. Para ello, nos hemos basado en una arquitectura CLIENTE - SERVIDOR en la que el primero se encarga de enviar los datos topológicos de la red y el segundo de optimizarla.
El cliente consta de dos aplicaciones. La primera de ellas, crea el esquema de la base de datos seleccionada, para que en él se decida qué campos de ésta se desean encriptar y cuáles no. Además ofrece la posibilidad de comprobar cómo funcionan los distintos tipos de algoritmos simétricos ofrecidos encriptando y desencriptando la base de datos elegida. A continuación, ejecutará la segunda aplicación. Ésta consta de una interfaz donde se fijarán los parámetros evolutivos que necesitará el Servidor. Con la información proporcionada, se codificará la información considerada crítica por el usuario y se enviará al Servidor Web
En primer lugar, el Servicio Web desencriptará los datos y ejecutará la optimización siguiendo el paradigma de la programación evolutiva. Resuelto lo anterior, volverá a codificar la información confidencial y transmitirá al Cliente la mejor solución encontrada.
Por último, el Cliente recibirá la optimización de la red de transporte y, tras desencriptarla, actualizará su base de datos.

The project is a Web Service in .NET that optimizes a transport net. We have used a CUSTOMER - SERVICE architecture where the first one sends the topologic information of the net and the second one optimizes it.
The Customer has two interfaces. One of them, is used to create a table in the selectionated database where the user will decided which fields are important to be encrypted. The other one, sets some parameters that the Service will need. Later, the Customer will encrypt the critical information and will send it to the Web Service.
Firstly, the Web Service will decrypt the information and will execute the optimization based in the evolutive programming model. Later, the Web Service will encrypt the confidential data again and will send the best found solution to the Customer.
Finally, the Customer will receive the optimization of the transport net and will update the database.

3. Lista de palabras clave.

1. .NET
2. SERVICIO
3. WEB
4. C#
5. OPTIMIZACIÓN
6. PROGRAMACIÓN
7. EVOLUTIVA
8. ENCRIPTAR
9. SEGURIDAD
10. XML

4. Introducción.

Los Servicios Web permiten hoy en día la comunicación entre un Cliente y un Servidor mediante lenguaje XML a través de la infraestructura de Internet. Pero un problema crítico en estas transacciones es el de la *seguridad* de los datos que se transmiten (por ejemplo, consulta de datos bancarios…). De ahí la importancia de desarrollar un proyecto que reúna las ventajas de los Servicios Web con la seguridad en la transmisión.

La memoria se compone de varias partes. Primeramente tiene lugar una breve presentación de los autores. En segundo término, hemos creído conveniente explicar algunos de los conceptos fundamentales para una buena comprensión del desarrollo del proyecto.

Seguidamente, en la sección 7, abordaremos de modo claro y modular el núcleo de la memoria: el desarrollo del proyecto completo.

Desde un punto de vista pedagógico hemos creído muy interesante explicar dos partes bien diferenciadas del proyecto: La *optimización* y la *seguridad*. Es en los siguientes capítulos donde explicamos con detalle dichas partes incluyendo aplicaciones prácticas.

Los más curiosos pueden encontrar mucha más información en los apéndices que tratan sobre la criptografía, la programación evolutiva y sobre el mundo XML.

Nos ha parecido interesante incluir al final de la memoria un glosario en donde poder consultar todos los términos que el lector considere necesario.

Finalmente, cerramos la memoria con una recopilación de fuentes de información en donde ampliar conocimientos en alguno de los aspectos relacionados con el proyecto.

5. ¿Quiénes somos?

Los responsables de que saliera este proyecto adelante somos tres compañeros y amigos de 5º de informática, ilusionados por aprender nuevas tecnologías y métodos de trabajo y siempre atentos a los consejos de nuestro director de proyecto, José Jaime Ruz Ortiz.

Aquí nos tenéis ☺

Abel Miguel Ibáñez Mojarro

Náyade Bermúdez Brito

Javier González del Castillo

6. Algunos conceptos previos.

Veamos algunos conceptos relacionados con el proyecto que son importantes tenerlos claros previamente:

6.1 ¿Qué es un Servicio Web?

Es un servicio, con un interfaz definido y conocido, al que se puede acceder a través de Internet. Igual que una página web está definida por un URL (Uniform Resource Locator), un servicio web está definido por un URI (Uniform Resource Identification) y por su interfaz, a través del cual se puede acceder a él.

Al igual que una página web puede ofrecer cotizaciones de la bolsa en forma visual, un servicio web que haga lo mismo presentará una interfaz para que se pueda acceder fácilmente a los mismos datos desde una aplicación cliente. De esta forma, las aplicaciones se convierten en clientes que integran servicios web procedentes de diferentes proveedores.

Para acceder a la aplicación, se realizan llamadas remotas a métodos a través de HTTP que pueden hacer uso del protocolo SOAP (Simple Object Access Protocol). Existen entornos propietarios para sistemas distribuidos donde se pueden realizar este tipo de llamadas remotas de forma más eficiente (velocidad) pero de modo más complejo y sin la ventaja de la interoperabilidad. Esto ocurre por ejemplo con DCOM (Distributed Compontent Object Model) o CORBA. SOAP es un estándar basado en XML que simplifica enormemente y estandariza los accesos remotos.

Una de las grandes ventajas de los Servicios Web es que pueden ser consumidos desde cualquier sistema operativo o plataforma de programación. Es decir, los Servicios Web desarrollados en un entorno .NET podrían ser invocados, por ejemplo, desde un entorno Java.

Adicionalmente, los Servicios Web pueden describirse completamente utilizando el lenguaje WSDL (Web Service Description Languaje), lo que hace posible el descubrimiento dinámico de Servicios Web en tiempo de ejecución a través de UDDI. WSDL permite describir todos los métodos (conjuntamente con los tipos necesarios para llamar a esos métodos) utilizando XML con esquemas XML.

6.2 La plataforma .NET y C#.

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada plataforma .NET, y a los servicios antes comentados se les denomina Servicios Web.

Para crear aplicaciones para la plataforma .NET, tanto servicios Web como aplicaciones tradicionales (aplicaciones de consola, aplicaciones de ventanas, servicios de Windows NT, etc.), Microsoft ha publicado el denominado kit de desarrollo de software conocido como .NET Framework SDK, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y Visual Studio .NET, que permite hacer todo lo anterior desde una interfaz visual basada en ventanas. Ambas herramientas puede descargarse gratuitamente desde http://www.msdn.microsoft.com/net.

El concepto de Microsoft.NET también incluye al conjunto de nuevas aplicaciones que Microsoft y terceros han (o están) desarrollando para ser utilizadas en la plataforma .NET. Entre ellas podemos destacar aplicaciones desarrolladas por Microsoft tales como Windows.NET, Hailstorm, Visual Studio.NET, MSN.NET, Office.NET, y los nuevos servidores para empresas de Microsoft (SQL Server.NET, Exchange.NET, etc.)

C# (leído en inglés “C Sharp” y en español “C Almohadilla”) es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programar usando este lenguaje resulta mucho más sencillo e intuitivo que hacerlo con cualquier otro, ya que C# carece de elementos heredados (innecesarios en .NET). Por todo esto, se suele decir que C# es el lenguaje nativo de .NET.

La sintaxis y estructuración de C# es muy similar a la de C++, ya que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Además, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic.

Un lenguaje que hubiese sido ideal utilizar para estos menesteres es Java, pero debido a problemas con la empresa creadora del mismo -Sun-, Microsoft ha tenido que desarrollar un nuevo lenguaje que añadiese a las ya probadas

virtudes de Java las modificaciones que Microsoft tenía pensadas para mejorarlo aún más y hacerlo un lenguaje orientado al desarrollo de componentes.

En resumen, C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues Microsoft ha escrito la mayor parte de la BCL usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el .NET Framework SDK

6.3 El universal XML.

XML, (eXtensible Markup Lenguage) lenguaje extensible de etiquetas , no es un lenguaje de marcado como el lenguaje HTML sino que es un metalenguaje. Es decir, un lenguaje para definir lenguajes. Los elementos que lo componen pueden dar información sobre lo que contienen, no necesariamente sobre su estructura física o presentación, como ocurre en HTML.

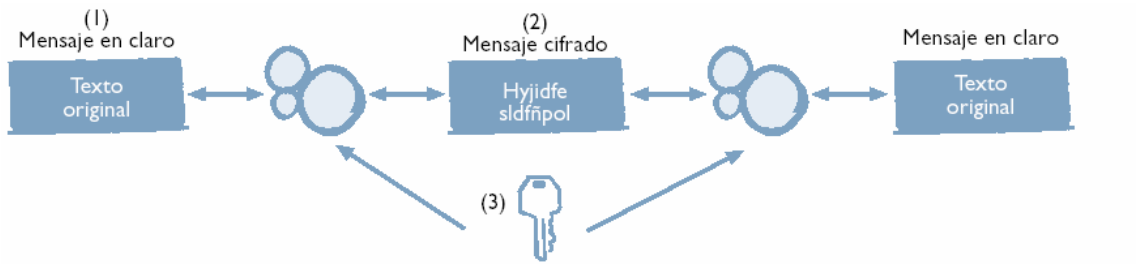
No ha nacido sólo para su aplicación en Internet, sino que se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, y casi cualquier cosa que podamos pensar.

En el entorno .NET juega un papel fundamental ya que es el lenguaje utilizado para los ficheros de configuración y la documentación del código fuente así como en SOAP, los Servicios Web y ADO.net, por citar algunas áreas.

6.4 La importancia de la seguridad.

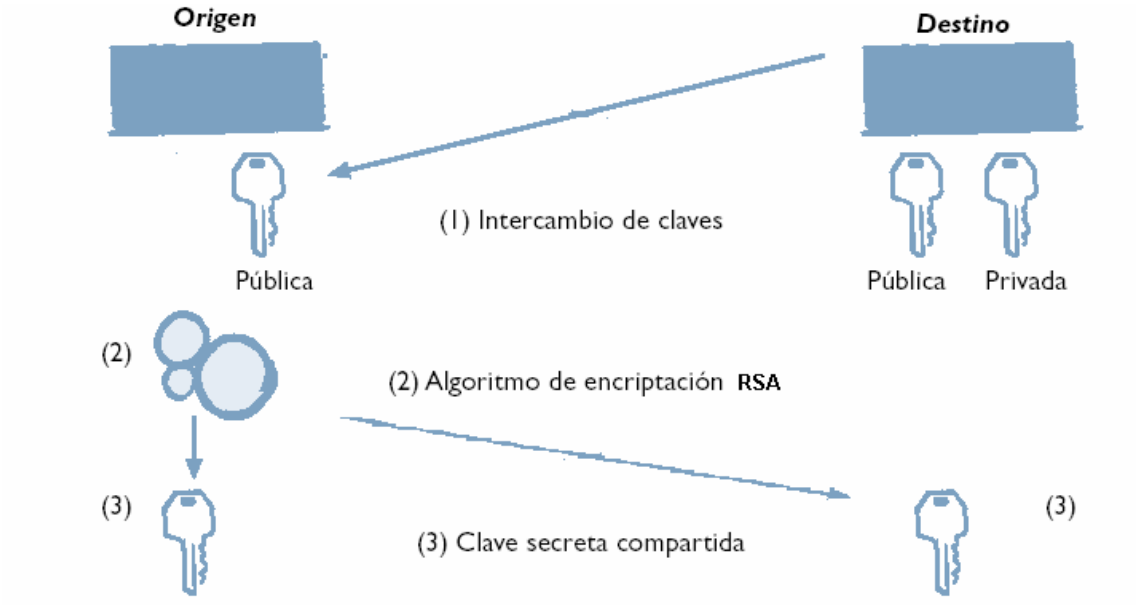
En el momento que hacemos uso de un Servicio Web, nuestros datos viajan por Internet hasta el servidor. En el camino, habilidosos personajes pueden interceptar información crítica (por ejemplo, datos bancarios...). Por eso, es imprescindible añadir una capa de seguridad que proteja los datos importantes.

Los algoritmos simétricos se suelen llamar ‘de clave secreta’. Los usuarios que intercambian información, tendrán esta clave guardada anteriormente, debe ser conocida por ambas partes de la comunicación antes de realizar el encriptado, ya que un mensaje cifrado con esta llave, sólo podrá ser descifrado con la misma llave. La principal ventaja de este tipo de algoritmos es su velocidad de operación. Sin embargo, su mayor vulnerabilidad es la necesidad de transmitirse la clave de alguna forma para que ésta no sea interceptada.
Una forma de solucionar este problema, es derivando la misma clave secreta a partir de una misma cadena (contraseña). Otra solución, es mandar de forma segura la llave de un extremo a otro. Esto se hace usando otro tipo de encriptación: Algoritmos Asimétricos.



(1) El mensaje de texto en claro se encripta usando la clave compartida.
(2) El paquete encriptado pasa a través de la red pública.
(3) En el destino, el mensaje es desencriptado con la misma clave compartida.

Por otra parte, el tipo de encriptación asimétrica, soluciona la desventaja anterior mediante el uso de una clave pública y otra privada. Este mecanismo tiene peor rendimiento para el cifrado de mensajes largos. Sin embargo, la distribución de claves es mucho más sencilla y no tiene problemas de seguridad.



Una solución muy extendida es usar una encriptación asimétrica para la transmisión inicial de la clave (simétrica) y posteriormente continuar la transmisión de forma simétrica (con la clave conocida por ambas partes) ganando en velocidad como hemos comentado antes. Éste modelo es el que sigue nuestro proyecto

6.5 ¿Qué es la Programación Evolutiva?

Es un nuevo paradigma dentro de la Inteligencia Artificial que interpreta la evolución natural como un proceso de aprendizaje e intenta reproducir ese mecanismo aplicándolo a la resolución de problemas complejos de búsqueda y optimización. Algunos de los problemas en los que se centran son:

- **Optimación de funciones**. En casos de funciones complejas, con múltiples óptimos y resistentes a las técnicas tradicionales; por ejemplo: los problemas de diseño, más complejos y menos definidos que los de análisis.
- Lo que se conoce como **Vida Artificial**. Simulación de sistemas dinámicos complejos. Aquí es donde radica su interés para la investigación sociológica.

La programación evolutiva fue propuesta en la década de 1960 y su creador fue L. J. Fogel Este desarrollo comenzó como un esfuerzo encaminado a crear inteligencia artificial basado en la evolución de máquinas de estado finitas.

Las estrategias evolutivas fueron propuestas por Ingo Rechenberg y Hans-Paul Schwefel en la década de 1970. Su principal objetivo era el de resolver problemas de optimización.

De manera general la computación evolutiva toma como base las ideas de la evolución propuestas por Charles Darwin y en los descubrimientos realizados por Gregor Mendel en el campo de la genética.

Nuestro optimizador sigue un modelo evolutivo para calcular el mejor resultado.

7. El Proyecto.

En esta sección expondremos de forma detallada y clara el desarrollo completo del proyecto. Por otra parte, hemos creído muy interesante desde el punto de vista pedagógico ilustrar 2 partes bien diferenciadas del proyecto: la optimización y la seguridad. Puedes encontrar toda la información al respecto en las secciones 8 y 9.

7.1 Desarrollo.

7.1.1 Tecnologías utilizadas.

El entorno de desarrollo ha sido la plataforma .NET con su lenguaje específico C# para lo cual hemos utilizado el MS Visual Studio .NET.

Las bases de datos son de tipo relacional y por compatibilidad con lo anterior, hemos utilizado MS ACCESS.

La transmisión de los datos ha sido mediante XML a través de los protocolos estándar de Internet.

Para la encriptación hemos hecho uso de las librerías de C# que implementan algunas funciones para el tratamiento general de algoritmos simétricos, asimétricos,... La codificación de los datos ha sido a nivel del contenido de las etiquetas XML.

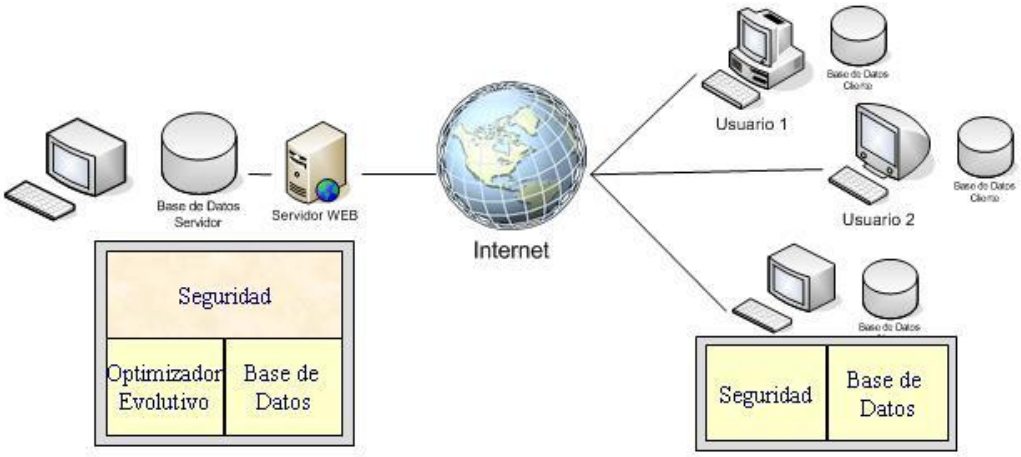
El optimizador al que llama el Servidor es un algoritmo basado en el modelo de la Programación Evolutiva desarrollado específicamente para este proyecto.

7.1.2 Arquitectura Cliente-Servidor.

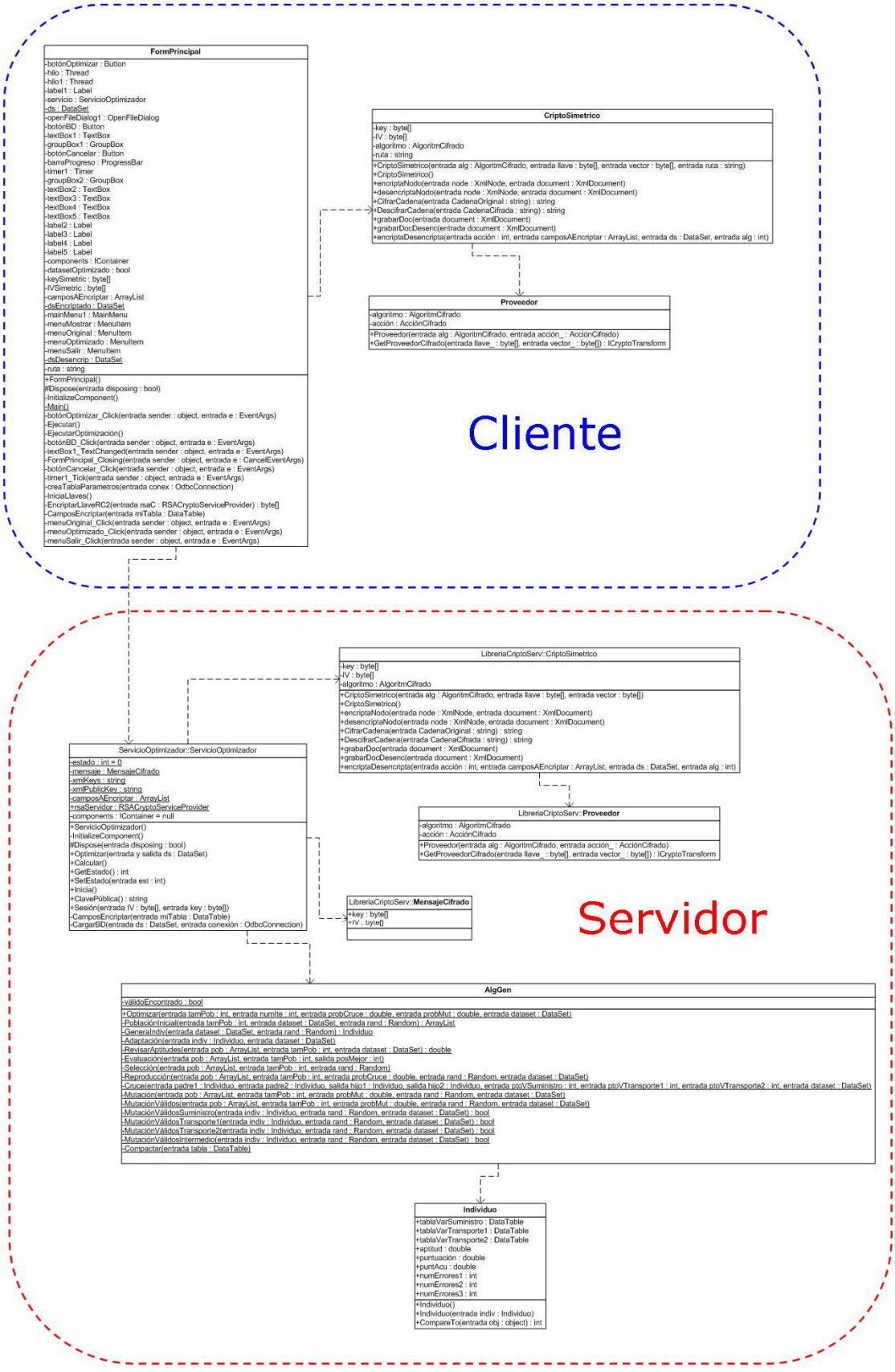
Para el desarrollo del proyecto hemos implementado una arquitectura CLIENTE - SERVIDOR ya que es la más extendida a la hora de implementar un Servicio Web. Esto es así porque justamente se adecua perfectamente a la naturaleza de un Servicio Web: un cliente solicita un "servicio" y un servidor se lo sirve.

Este modelo, además, permite la modularización total de los componentes clarificando el diseño e implementación del proyecto.

ESQUEMA DE LA ARQUITECTURA



7.1.3 Diagrama de clases.



7.1.4 El cliente pide de forma segura.

La aplicación cliente ofrece dos interfaces. Una de ellas servirá para que el usuario cree un esquema de la base de datos en el que se encontrarán todas las tablas que esta contenga, sus columnas y una opción que estará a ‘1’ si ese campo se desea encriptar.

La segunda interfaz servirá para determinar los valores referentes al algoritmo genético que necesitará el Servicio Web. Una vez fijados, el cliente y el servidor "dialogan" para establecer la clave de encriptación. Para ello, mediante el algoritmo RSA, el servidor mandará su clave pública al cliente. Éste generará la llave y el vector usando el algoritmo simétrico RC2. Una vez generadas las claves, encriptará la llave mediante la clave pública RSA y se las mandará de vuelta al servidor. Establecidas las claves a un lado y a otro de la comunicación, los datos son transmitidos de forma segura hacia el Servidor.

7.1.5 El servidor optimiza y devuelve el resultado.

El Servidor (Servicio Web) recibe los datos, los desencripta y llama al optimizador. Una vez que ha calculado el resultado de la petición del cliente, encripta nuevamente los datos sensibles siguiendo la tabla ‘Esquema’ y se los manda al Cliente.

7.1.6 El optimizador evolutivo.

El Servidor hace una llamada a un algoritmo genético que es el encargado de optimizar la red de transporte que le ha mandado el Cliente

7.1.7 ¿Cómo se transmite? DataSet + XML.

La transmisión de bases de datos se realiza mediante unas estructuras llamadas DataSet, parte fundamental del acceso a datos en Microsoft .NET Framework. Son objetos en memoria que pueden almacenar tablas, vistas y relaciones. C# aporta métodos para conectarse a una base de datos y volcar los datos a este tipo de estructuras de manera más o menos cómoda. Una vez descargados los datos, la conexión a la base de datos ya no se utilizará hasta que se desee cargar más datos o actualizar con los cambios realizados.

En última instancia, lo que se transmite es un documento XML con los datos del DataSet. En este nivel es donde se aplica la encriptación.

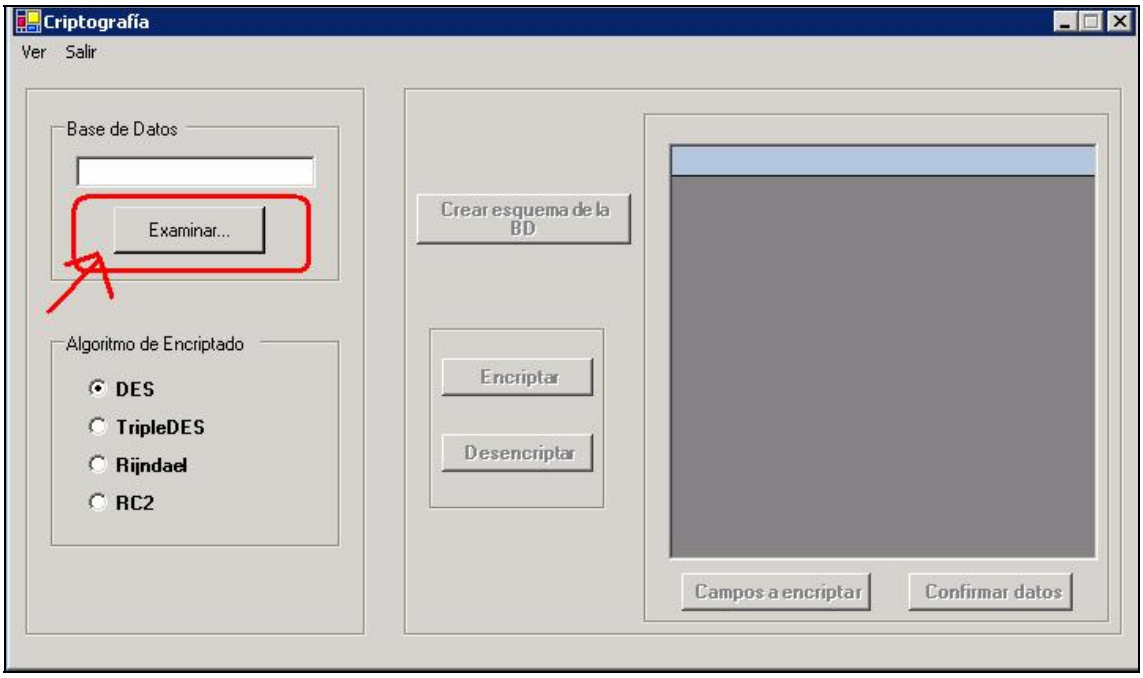
7.1.8 La capa de seguridad (algoritmos simétricos, asimétricos...).

Una parte fundamental en las transmisiones de datos por Internet es asegurar la confidencialidad de datos sensibles (como por ejemplo datos bancarios...). En nuestro proyecto, el Cliente tiene la posibilidad de determinar qué datos son críticos.
En un principio, habrá un diálogo con clave asimétrica, usando RSA, entre el Servidor y el Cliente. Tras este diálogo quedará determinada la clave simétrica que posteriormente utilizarán para la transmisión de los datos. Hemos fijado para la encriptación de datos el algoritmo simétrico RC2, aunque la aplicación está preparada para trabajar con cualquier tipo de algoritmo simétrico.
Ésta encriptación simétrica se realizará a nivel del documento XML que se genera para ser transmitido por la red.

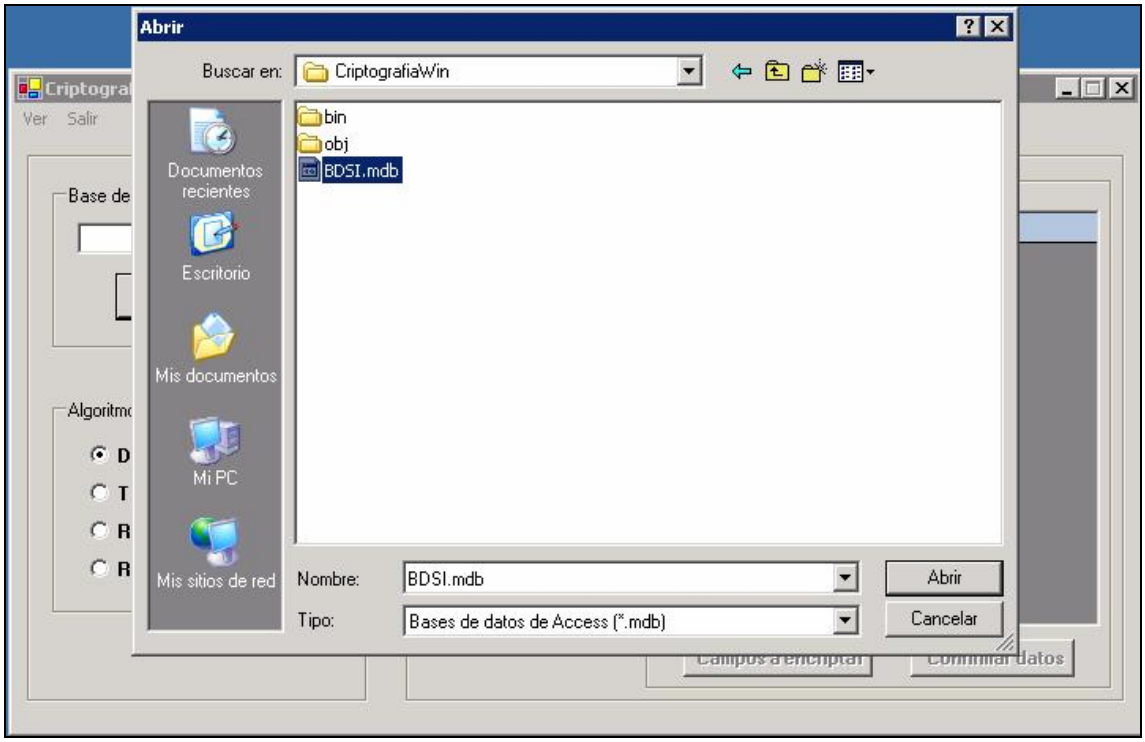
7.2 Guía de uso.

Veamos a continuación el desarrollo del proyecto completo comenzando con una guía rápida…

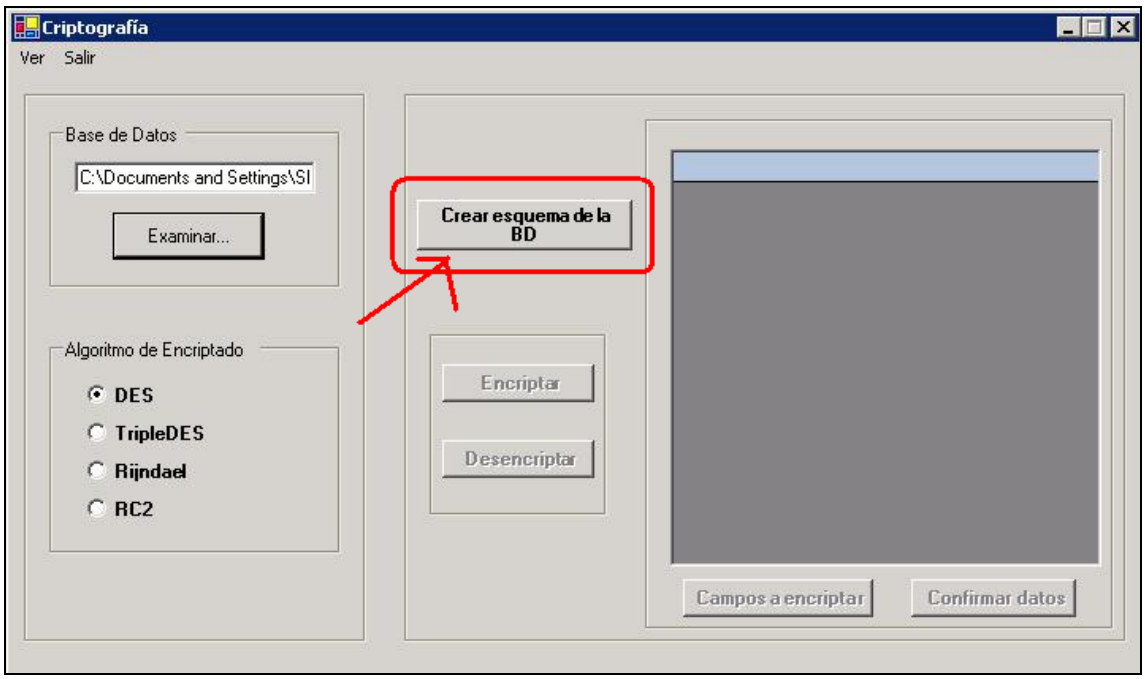
1. Ejecutamos la primera de las dos aplicaciones del cliente, y nos encontramos con la siguiente interfaz:



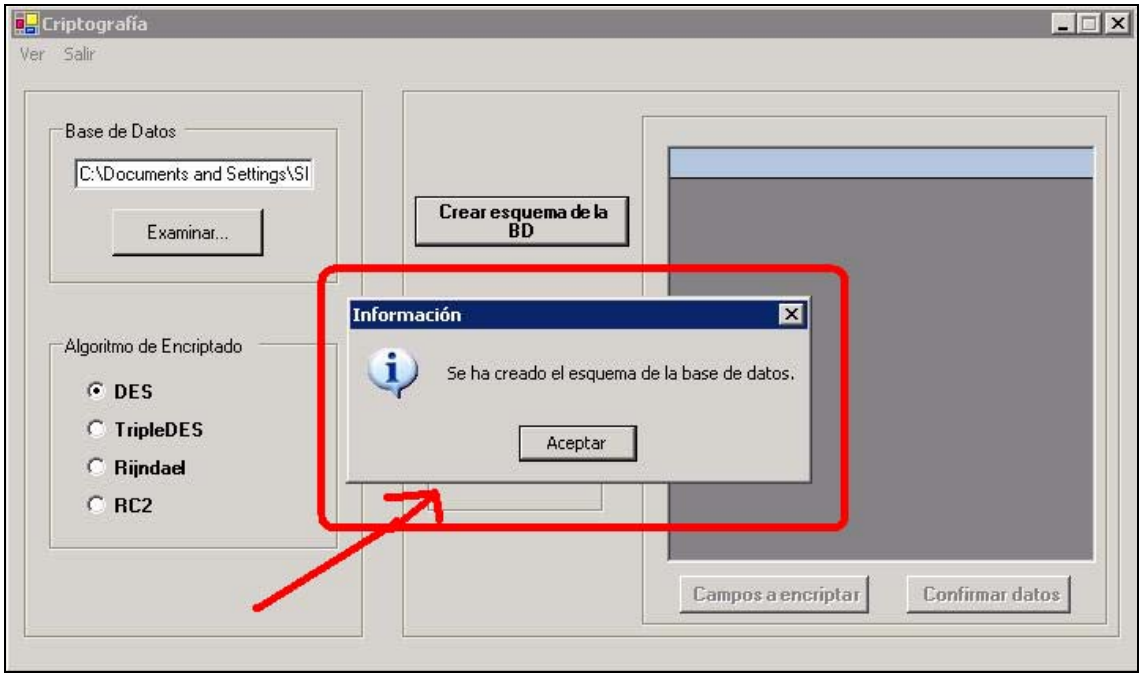
2. Seleccionamos la base de datos que queremos optimizar.



3. Habilitado el botón “Crea esquema de la BD”, pulsamos en él.



4. Si la base de datos es válida, nos avisa del éxito de la operación.

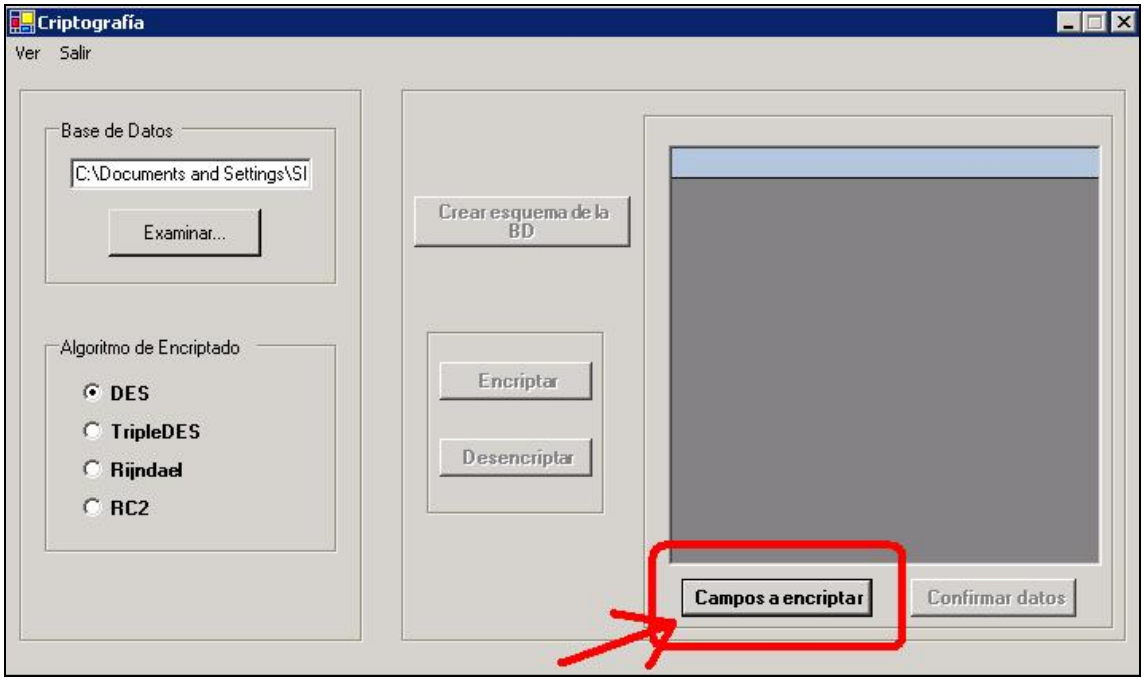


5. Ahora tenemos dos opciones:

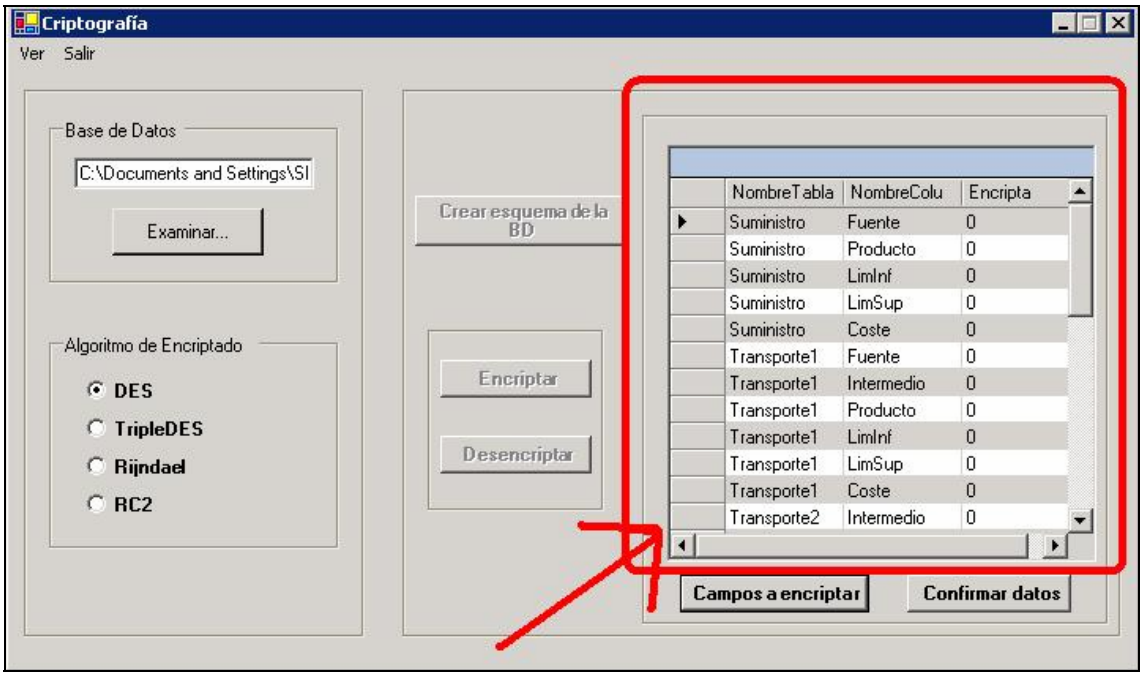
   5.1. Cerrar la aplicación, abrir la base de datos seleccionada con Access y desde ahí, modificar la tabla 'Esquema' señalando con 1 los campos que consideramos críticos.

   5.2 Modificar la tabla 'Esquema' desde la aplicación actual. Si hemos elegido esta opción, pasamos al punto 6.

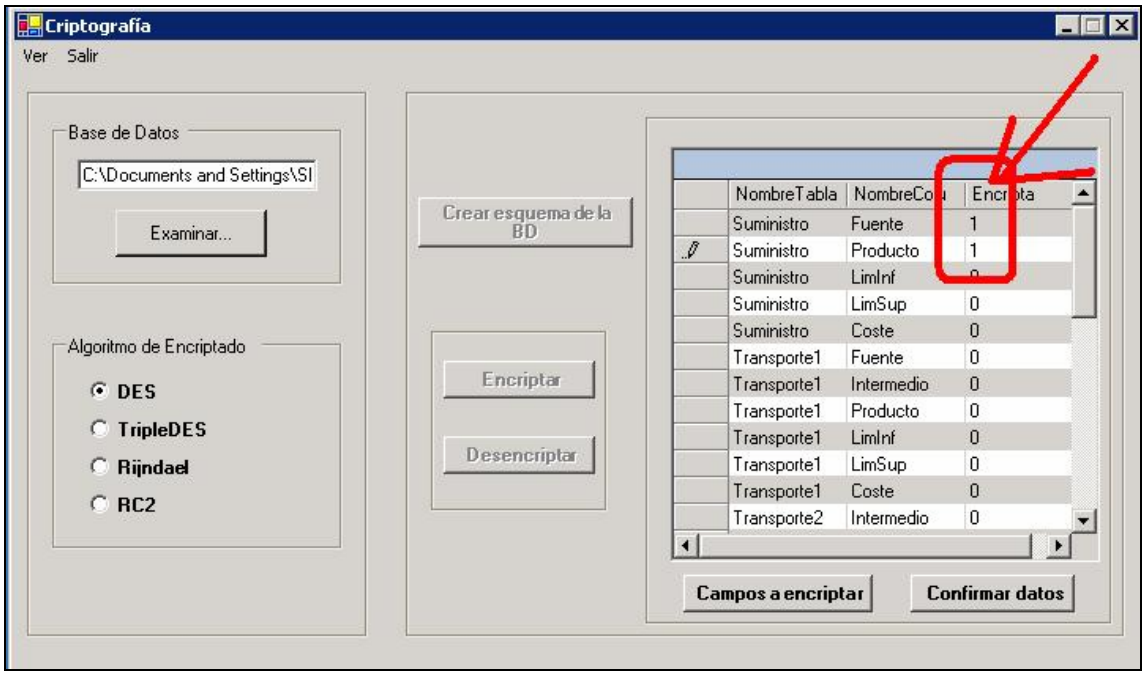
6. Pinchamos en el botón "Campos a encriptar".



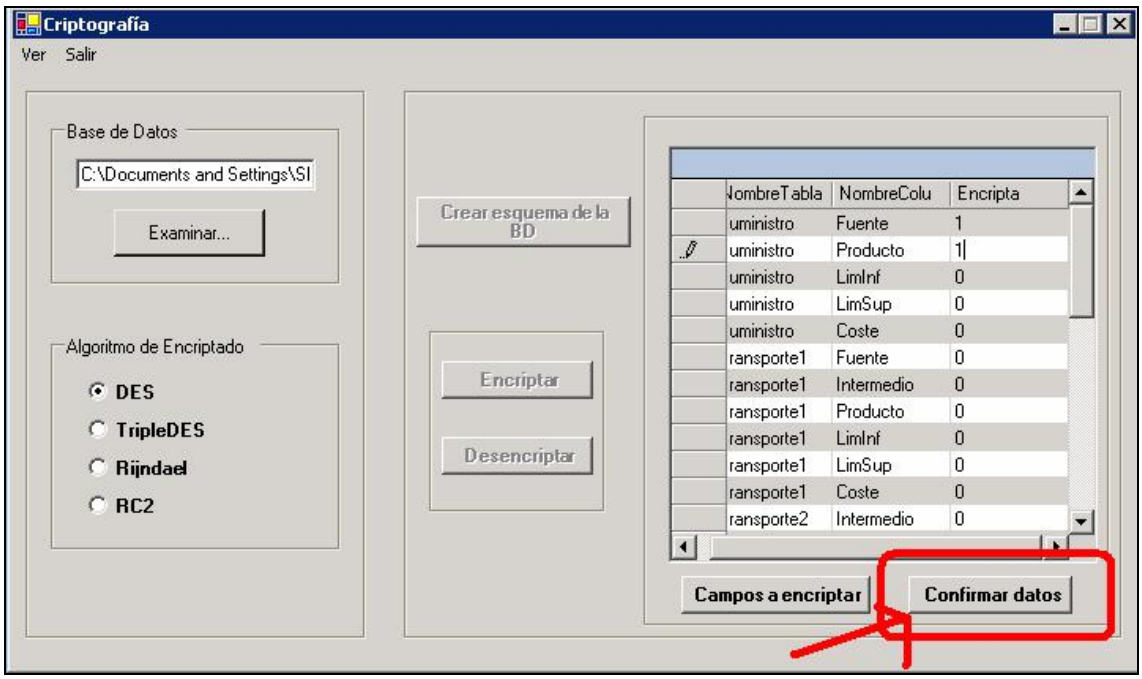
7. A continuación se nos muestra una tabla donde la primera columna corresponde a los nombres de las tablas existentes en la base de datos elegida; la segunda columna corresponde a los campos de cada tabla; y la tercera columna nos da la información de si deseamos encriptar o no ese campo. Por defecto, todos los campos aparecen como no críticos ('0').



8. Si queremos encriptar un campo determinado (por ejemplo las dos primeras filas), no tenemos más que poner un "1" en la última columna de la fila que queramos codificar



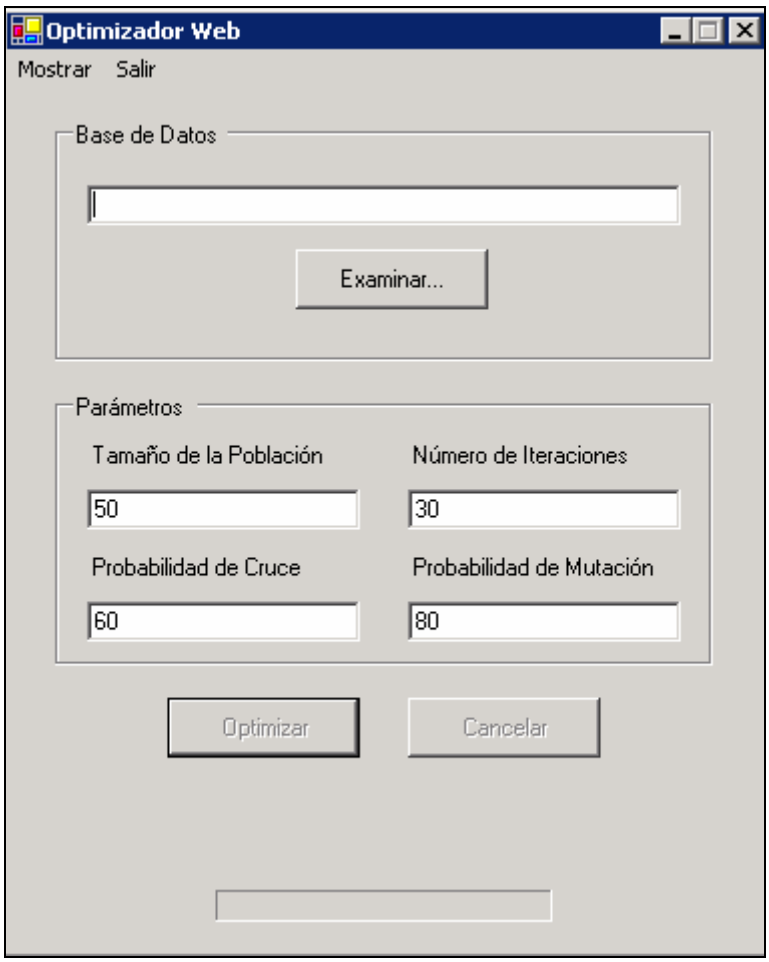
9. Cuando se haya decidido los campos críticos, pulsamos “Confirmar datos”.



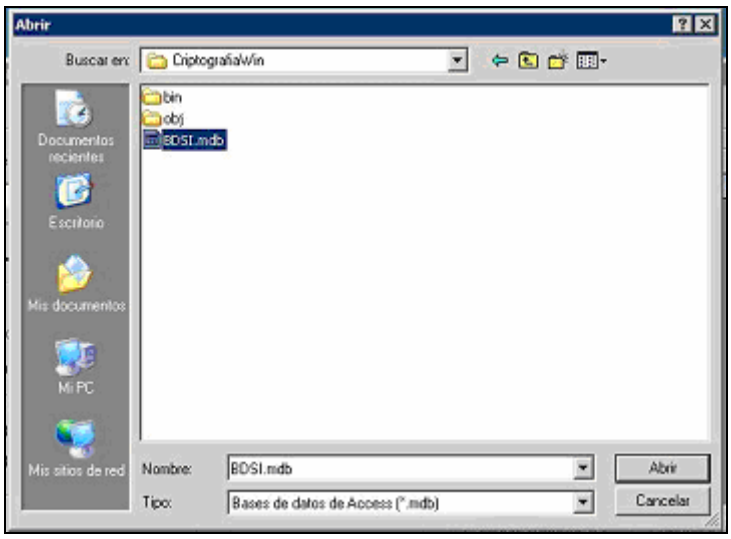
10. Ya podemos salir de la aplicación.

    Nota: Como podemos observar, en esta aplicación hay otras opciones como elegir el algoritmo simétrico de encriptado, encriptar, etc. Estas opciones nos dan una idea del aspecto con el que queda una determinada base de datos encriptada dependiendo del algoritmo de cifrado utilizado. Hay que señalar que para el uso del optimizador no es necesario seguir más que los pasos detallados hasta aquí. Para más información sobre otros usos de esta aplicación, ver punto 9.

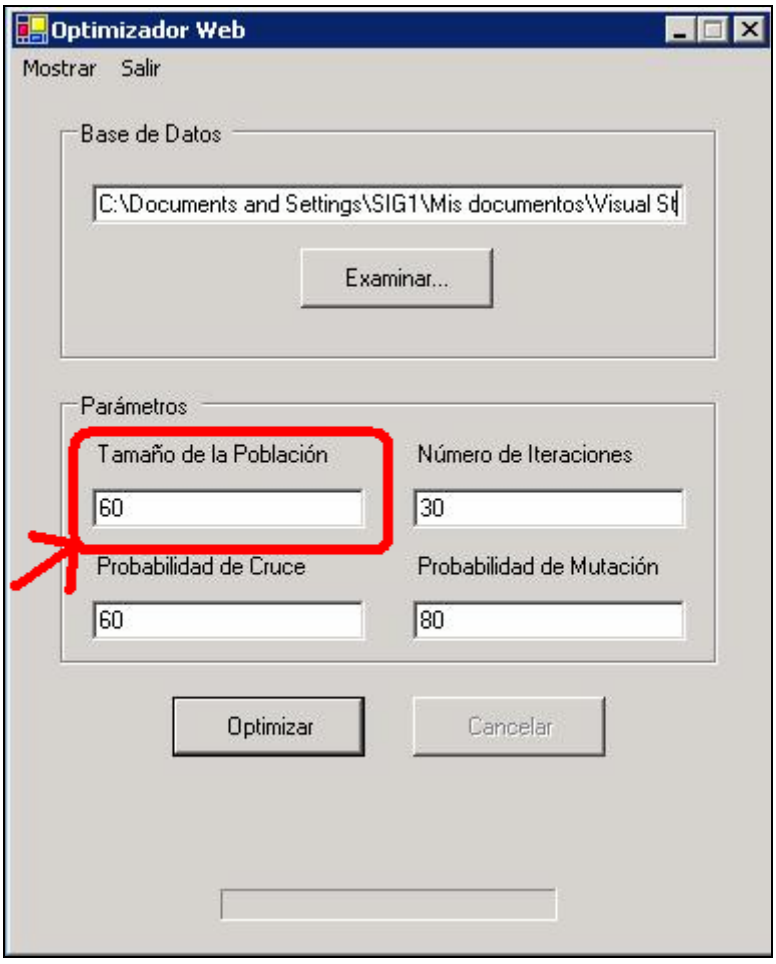
11. Abrimos la segunda interfaz del cliente, y nos encontramos con lo siguiente:



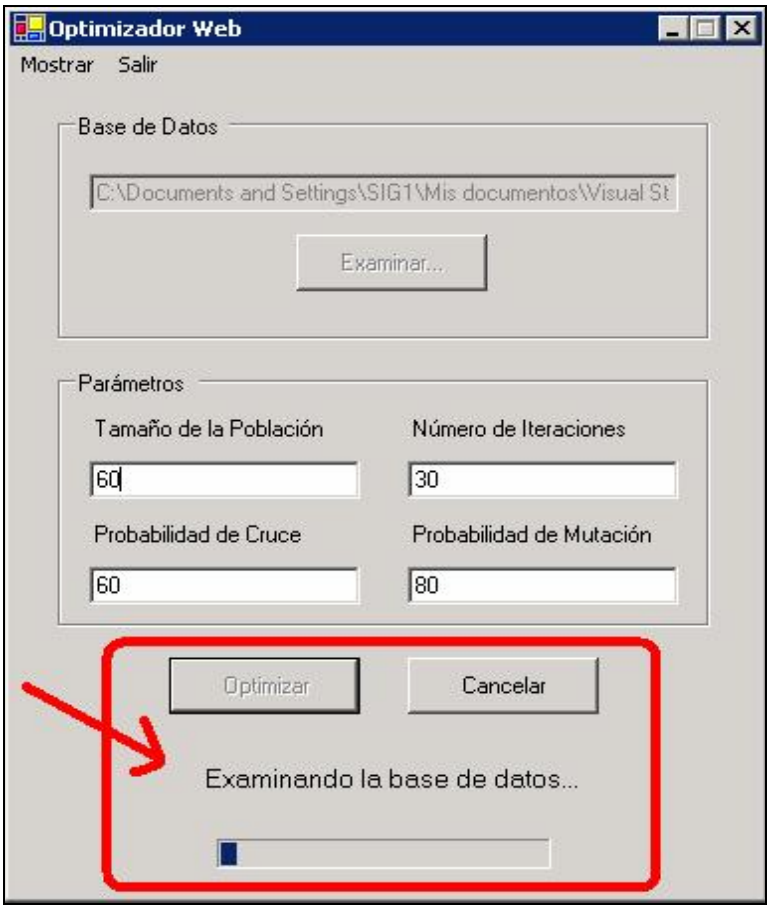
12. Seleccionamos la base de datos anterior, para la que hemos creado el esquema.

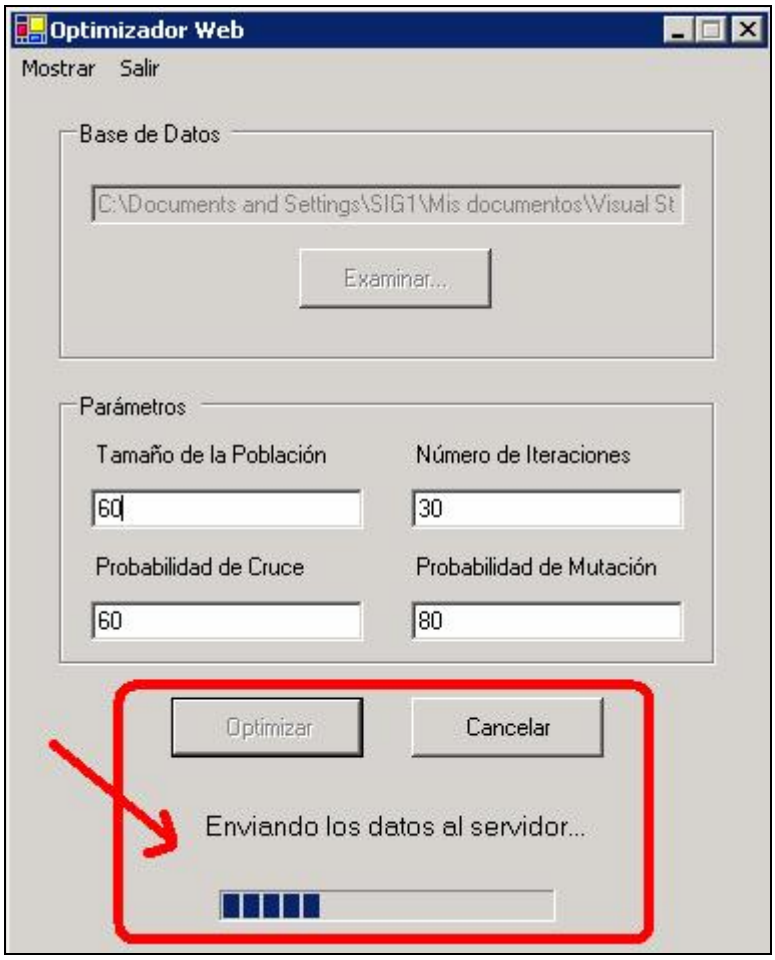


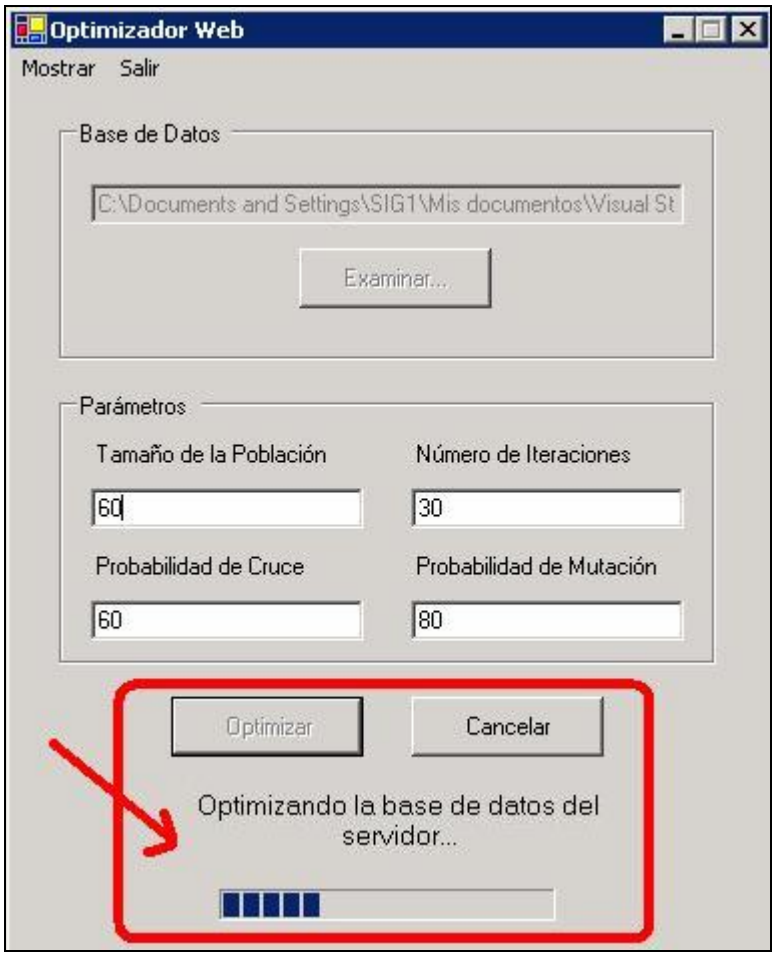
13. En la parte inferior de la aplicación podemos observar una ventana que se titula “Parámetros” con unos cuantos valores por defecto. Estos son los parámetros que necesitará el algoritmo genético para optimizar la red. Puedes modificarlos o bien dejar los que vienen. En nuestro ejemplo, vamos a aumentar en 10 el tamaño de la población, es decir, pondremos 60 en la parte de “Tamaño de la Población”:

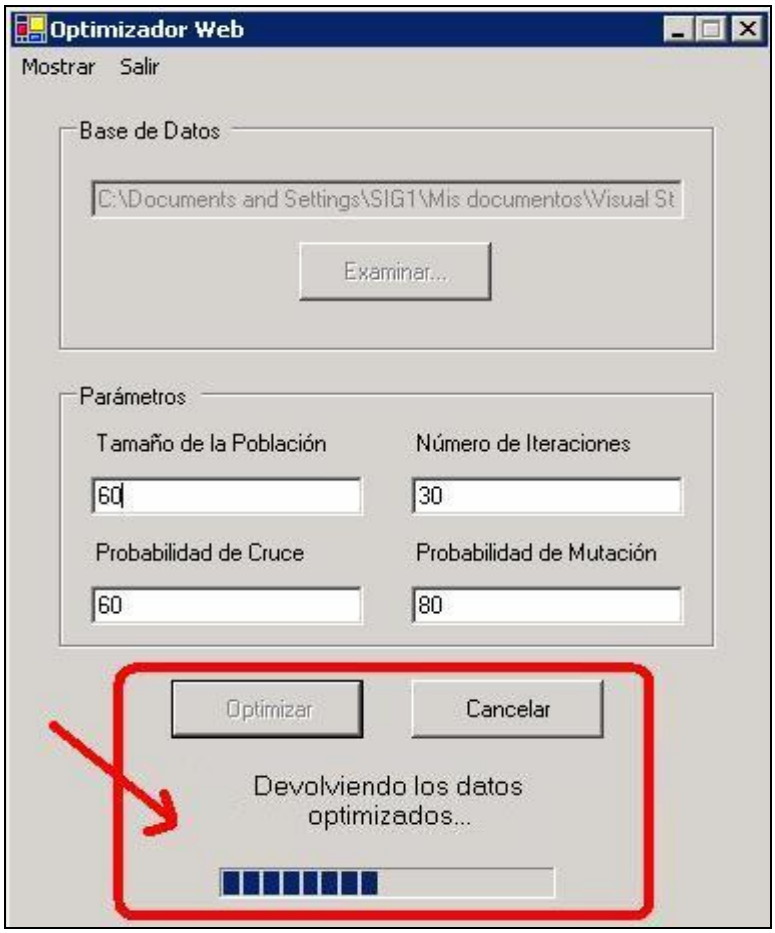


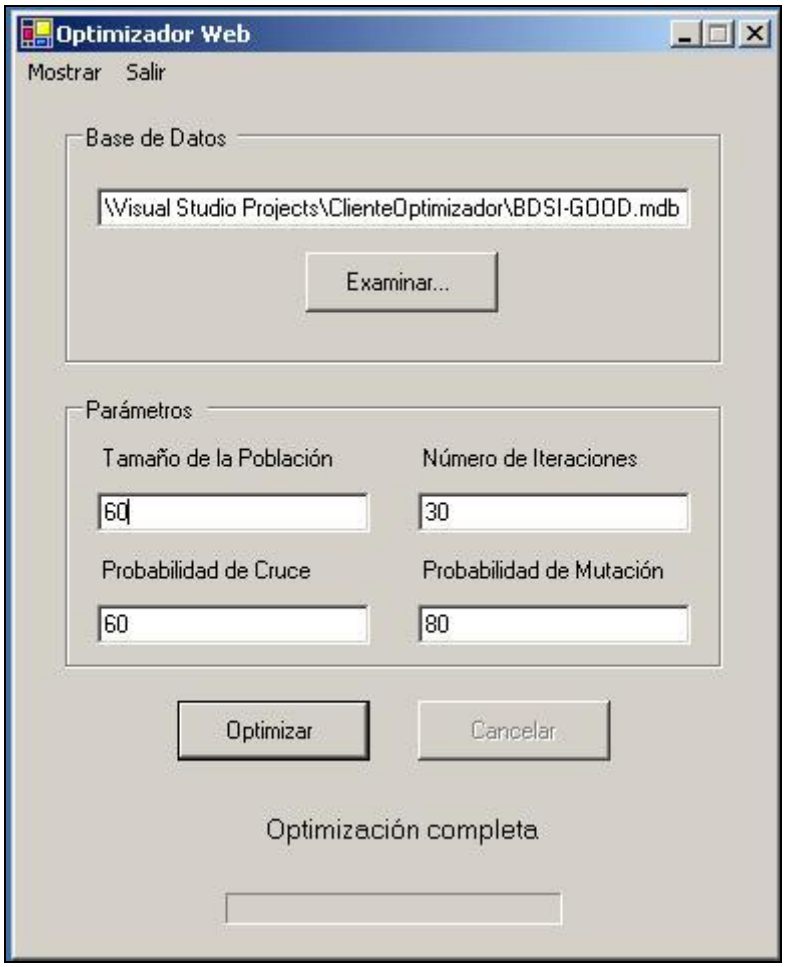
14. A continuación, la aplicación nos va mostrando la evolución del proceso.


Optimizador Web
Mostrar
Salir
Base de Datos
C:\Documents and Settings\SIG1\Mis documentos\Visual St
Examinar...
Parámetros
Tamaño de la Población
Número de Iteraciones
60
30
Probabilidad de Cruce
Probabilidad de Mutación
60
80
Optimizar
Cancelar
Examinando la base de datos...


Optimizador Web
Mostrar Salir
Base de Datos
C:\Documents and Settings\SIG1\Mis documentos\Visual St
Examinar...
Parámetros
Tamaño de la Población
Número de Iteraciones
60
30
Probabilidad de Cruce
Probabilidad de Mutación
60
80
Optimizar
Cancelar
Enviando los datos al servidor...

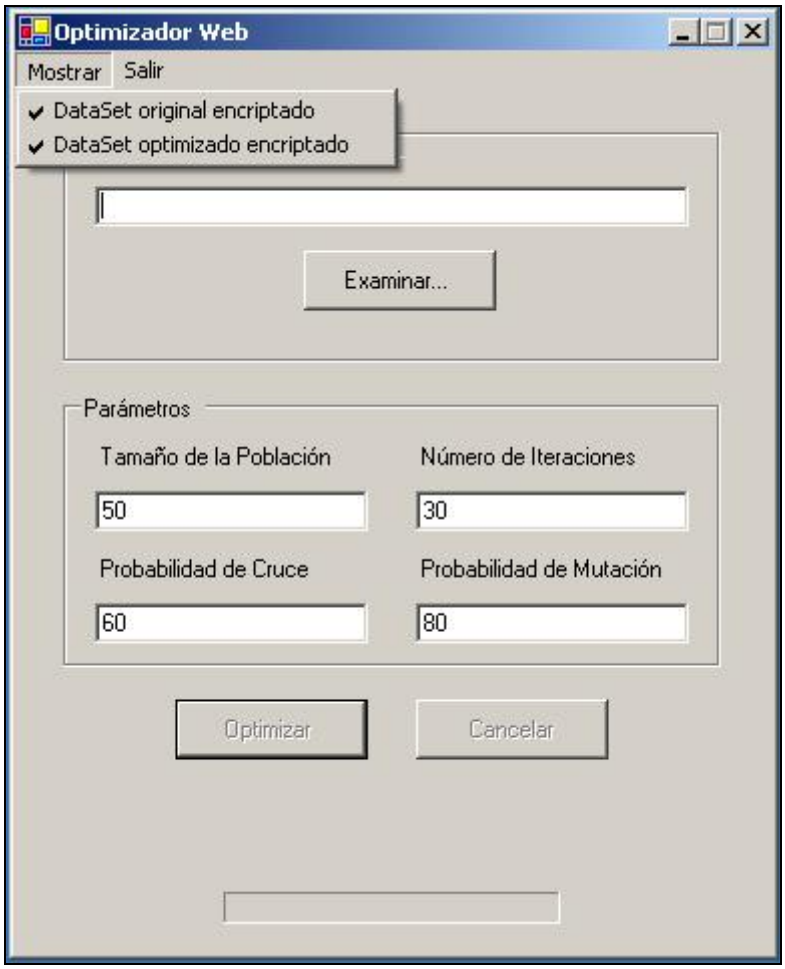

Optimizador Web
Mostrar Salir
Base de Datos
C:\Documents and Settings\SIG1\Mis documentos\Visual St
Examinar...
Parámetros
Tamaño de la Población
Número de Iteraciones
60
30
Probabilidad de Cruce
Probabilidad de Mutación
60
80
Optimizar
Cancelar
Optimizando la base de datos del servidor...


Optimizador Web
Mostrar
Salir
Base de Datos
C:\Documents and Settings\SIG1\Mis documentos\Visual St
Examinar...
Parámetros
Tamaño de la Población
60
Número de Iteraciones
30
Probabilidad de Cruce
60
Probabilidad de Mutación
80
Optimizar
Cancelar
Devolviendo los datos optimizados...


Optimizador Web
Mostrar
Salir
Base de Datos
\Visual Studio Projects\ClienteOptimizador\BDSI-GOOD.mdb
Examinar...
Parámetros
Tamaño de la Población
60
Número de Iteraciones
30
Probabilidad de Cruce
60
Probabilidad de Mutación
80
Optimizar
Cancelar
Optimización completa

15. La aplicación nos ofrece dos opciones:
    a. Ver la base de datos encriptada (como archivo XML) antes de ser mandada al servidor.
    b. Ver la base de datos encriptada cuando vuelve del servidor ya optimizada.

    Se pueden elegir cualquiera de las dos, o las dos, tal y como se muestra a continuación:

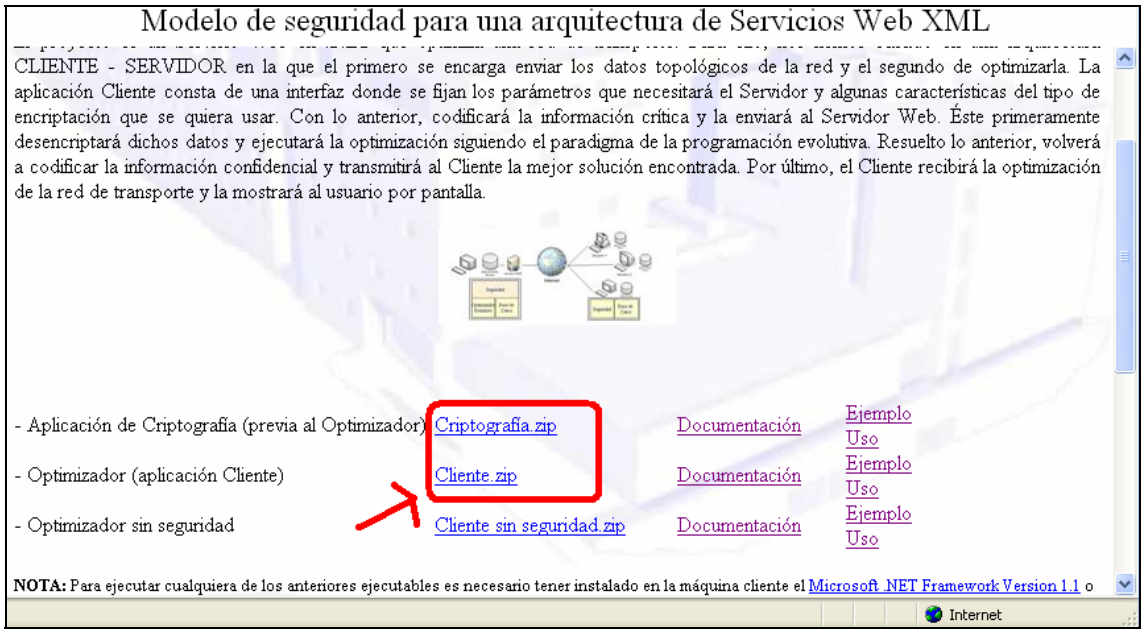


7.3 ¡Bájate el proyecto!

Puedes bajarte el proyecto en esta dirección:
http://www.fdi.ucm.es/datos/Docu_Docente2.asp

y en esta página pinchas en “Proyecto”

Se cargará una página donde aparece una breve introducción al proyecto y un poco más abajo aparecen unos enlaces para bajarse el proyecto. Deben bajarse los archivos “Criptografía.zip” y “Cliente.zip”:

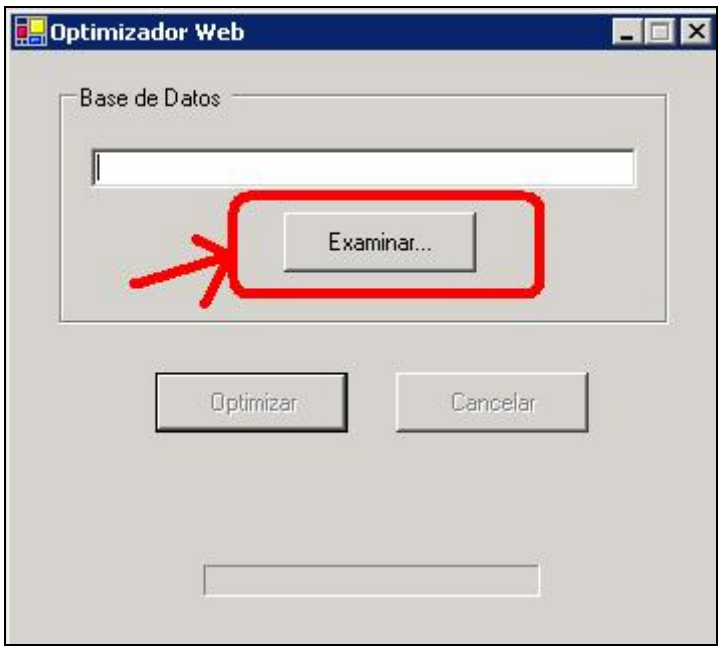


8. El optimizador sin seguridad.

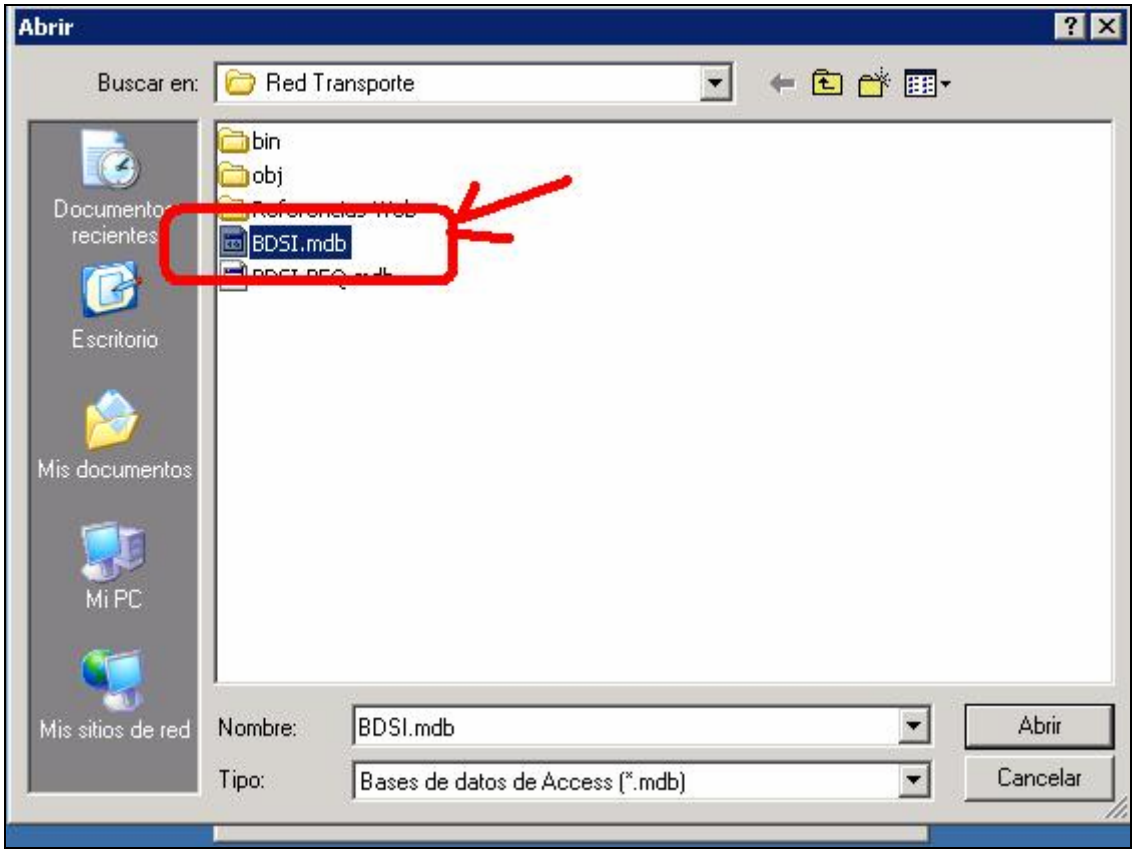
Para ilustrar de forma sencilla el funcionamiento del optimizador, hemos desarrollado una aplicación en la que se optimiza una base de datos seleccionada por el usuario. En esta aplicación, detalles de la programación evolutiva como son parámetros de tamaño de población, número de iteraciones, etc. no se dejan a elección del usuario. De la misma forma, en este caso, no se tienen en cuenta temas relacionados con la seguridad.

Veamos su funcionamiento:

1- Inicialmente pinchamos en el botón “examinar” para seleccionar la base de datos a optimizar:

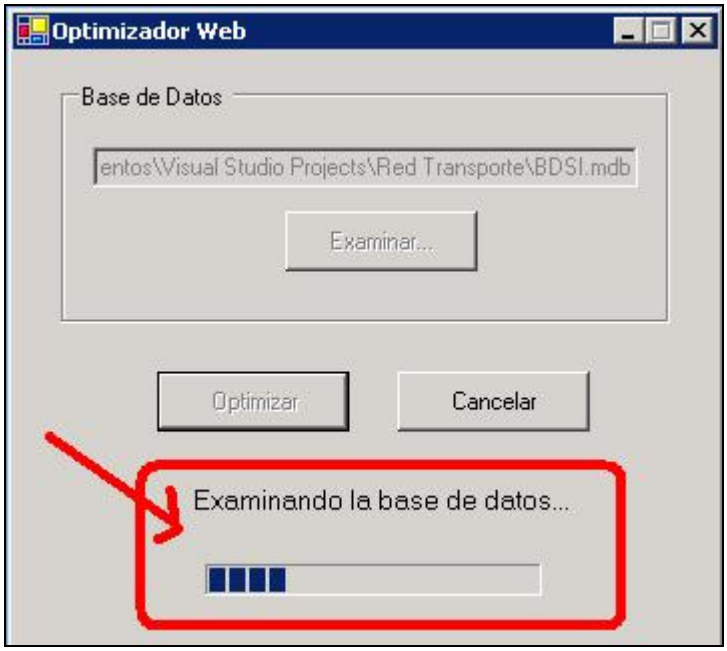


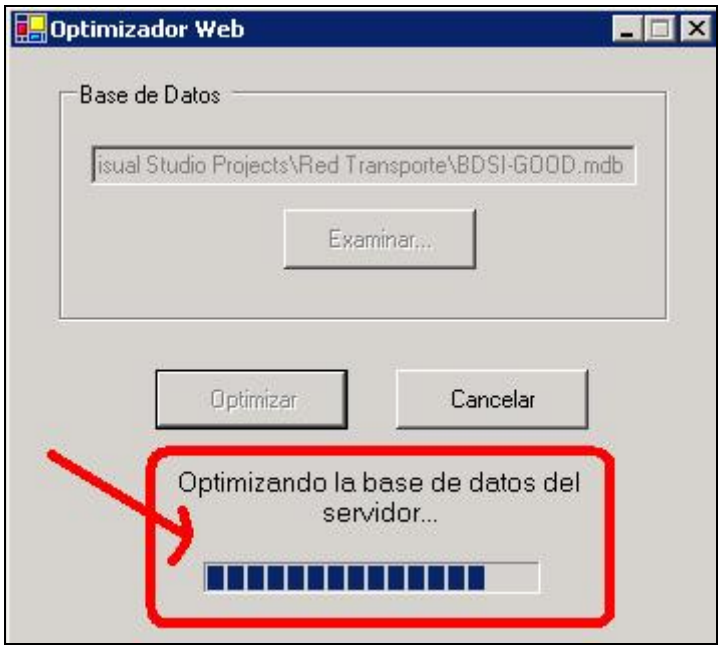
2- Seleccionamos la base de datos.

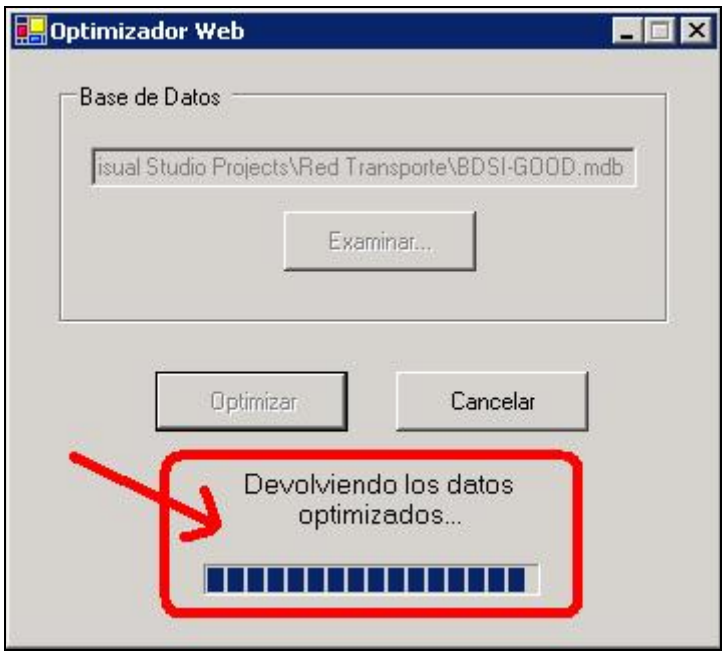


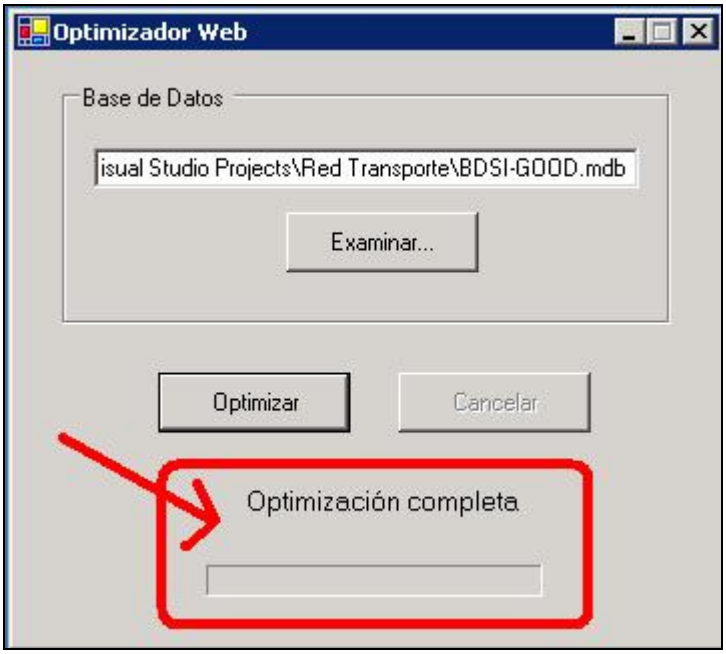
3- Se ha habilitado el botón “Optimizar”. Pinchamos sobre él.

4- A continuación, la aplicación nos va mostrando la progresión del proceso.









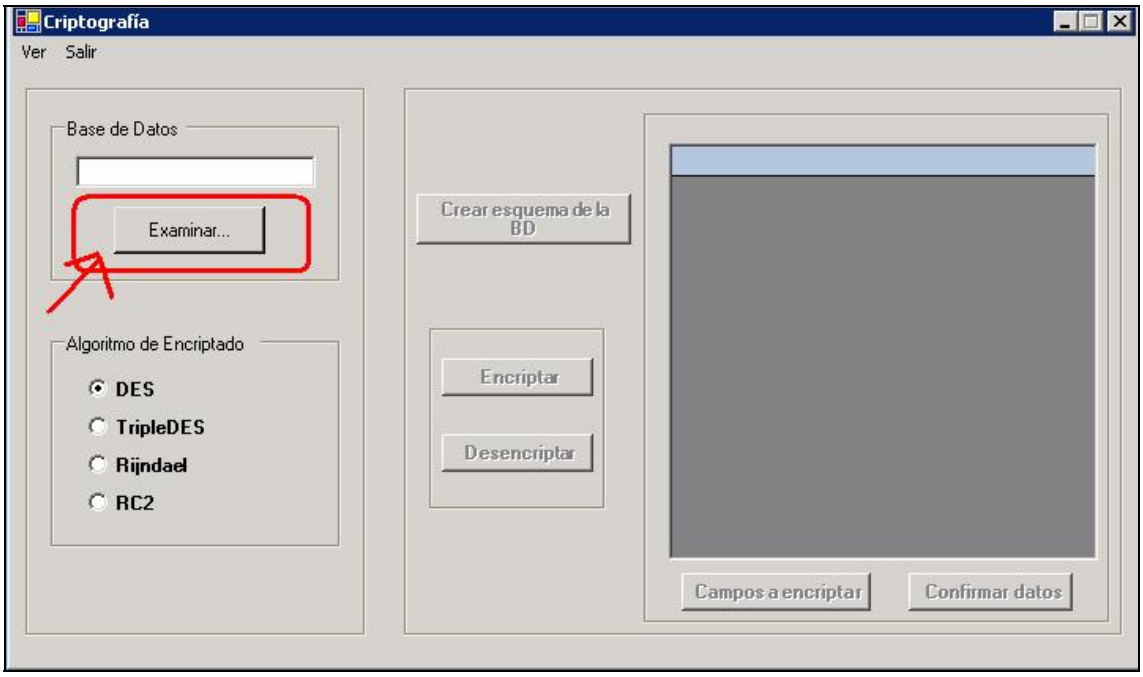
Ahora no tendríamos más que ir a la base de datos y comprobar la optimización resultante.

9. La seguridad.

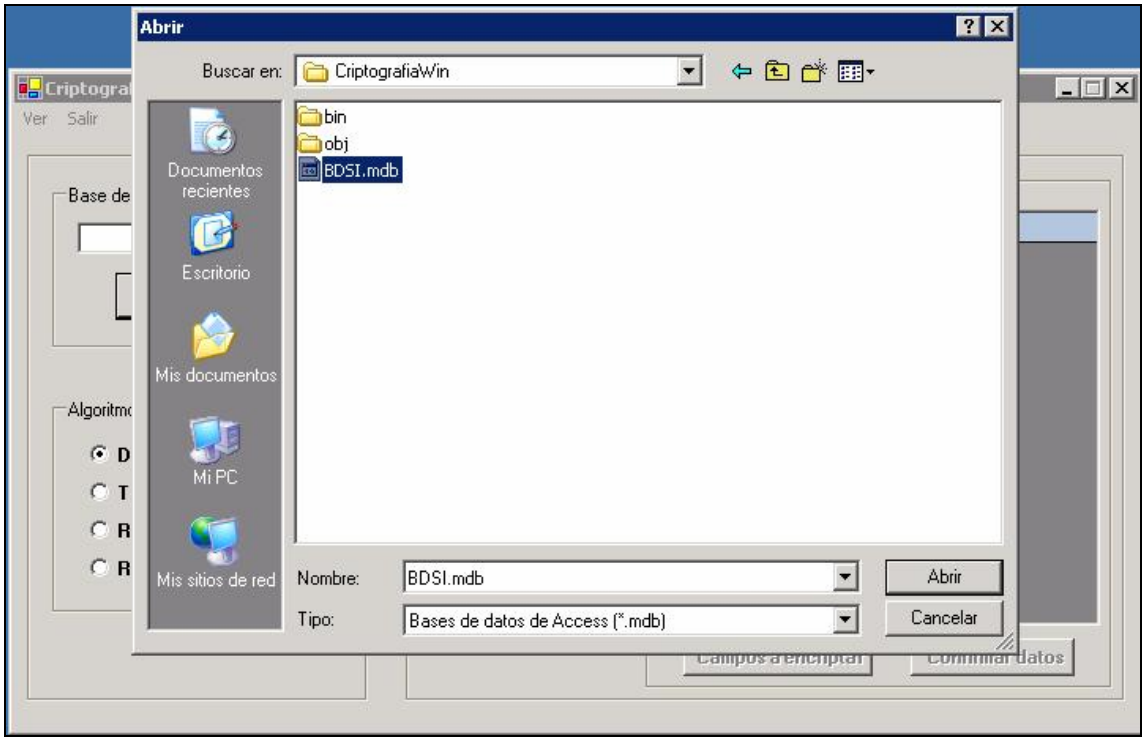
Como hemos dicho anteriormente, uno de los puntos fundamentales en el proyecto es la seguridad en la transmisión.
En el punto 7.1 hemos visto el uso básico de la siguiente aplicación. Éste era crear el esquema de la base de datos y seleccionar los campos que consideramos críticos para que sean encriptados. A continuación vamos a detallar otras funciones de esta aplicación.
Esta aplicación ofrece la posibilidad de ver los distintos aspectos de una base de datos encriptada según el tipo de algoritmo de cifrado simétrico elegido.

A continuación vemos un funcionamiento típico de la aplicación:

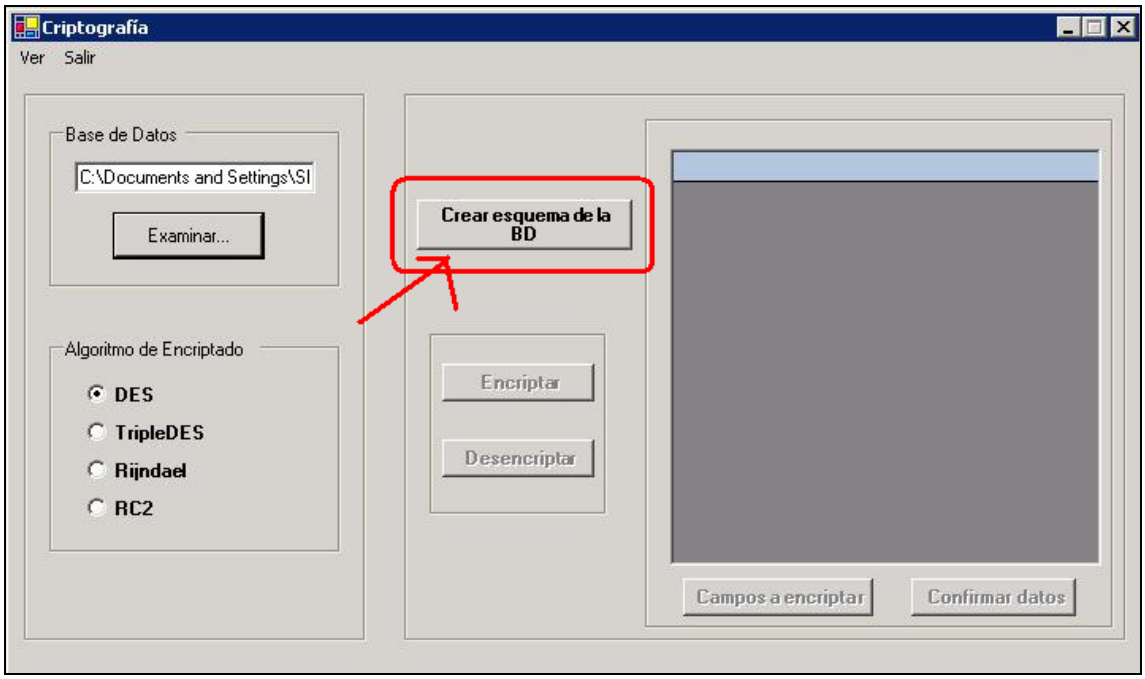
1. Interfaz inicial. Pinchamos en “examinar” para cargar una base de datos:



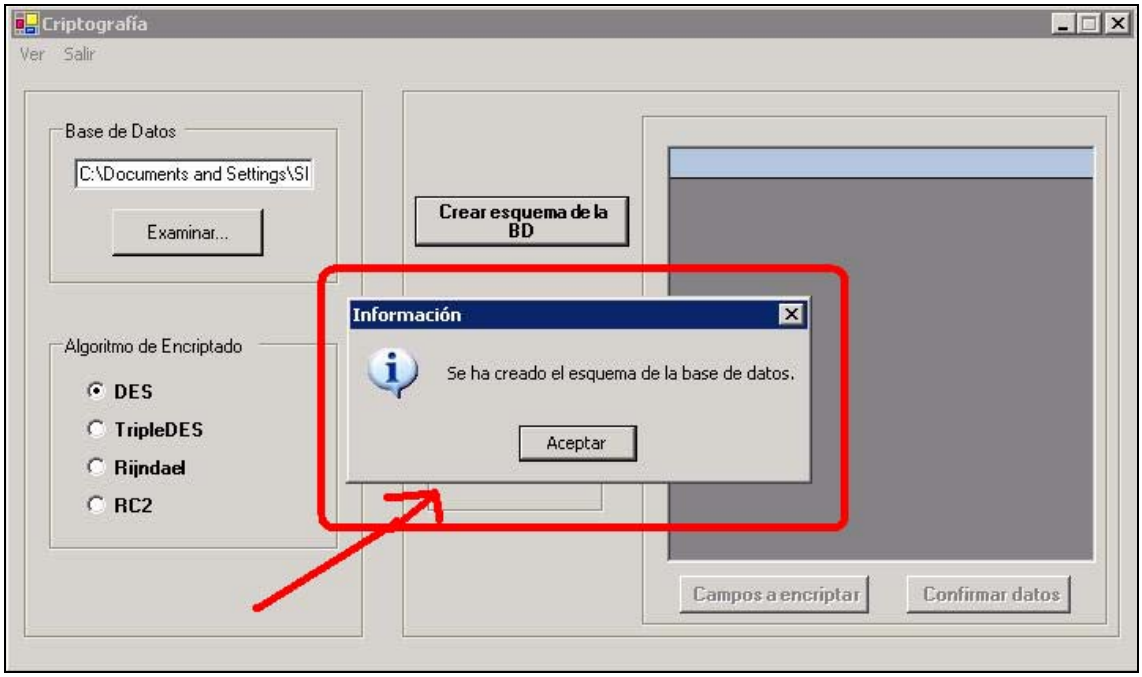
2. Seleccionamos una base de datos.



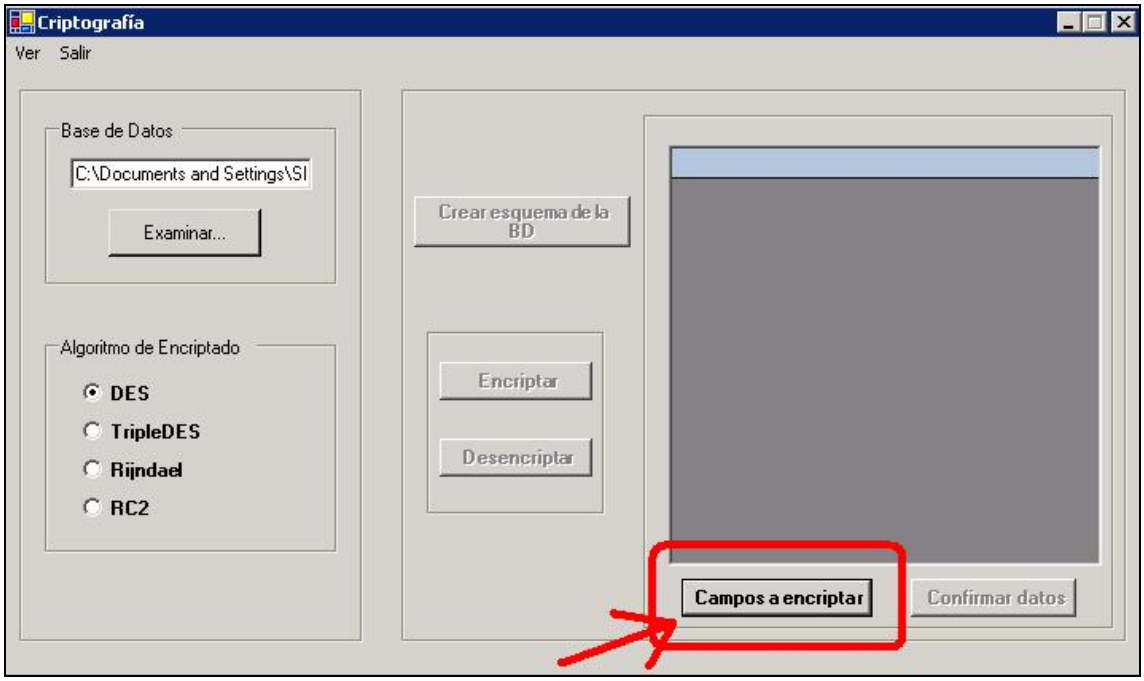
3. Creamos el esquema de la base de datos.



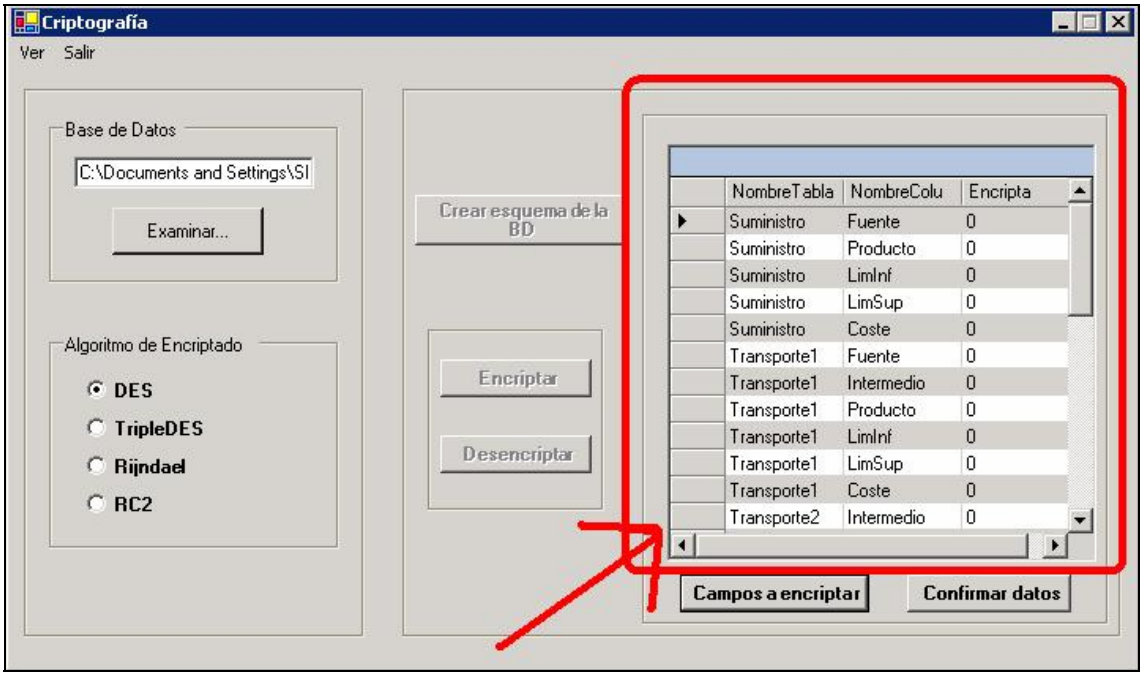
4. Un mensaje nos informará de que se ha creado correctamente.



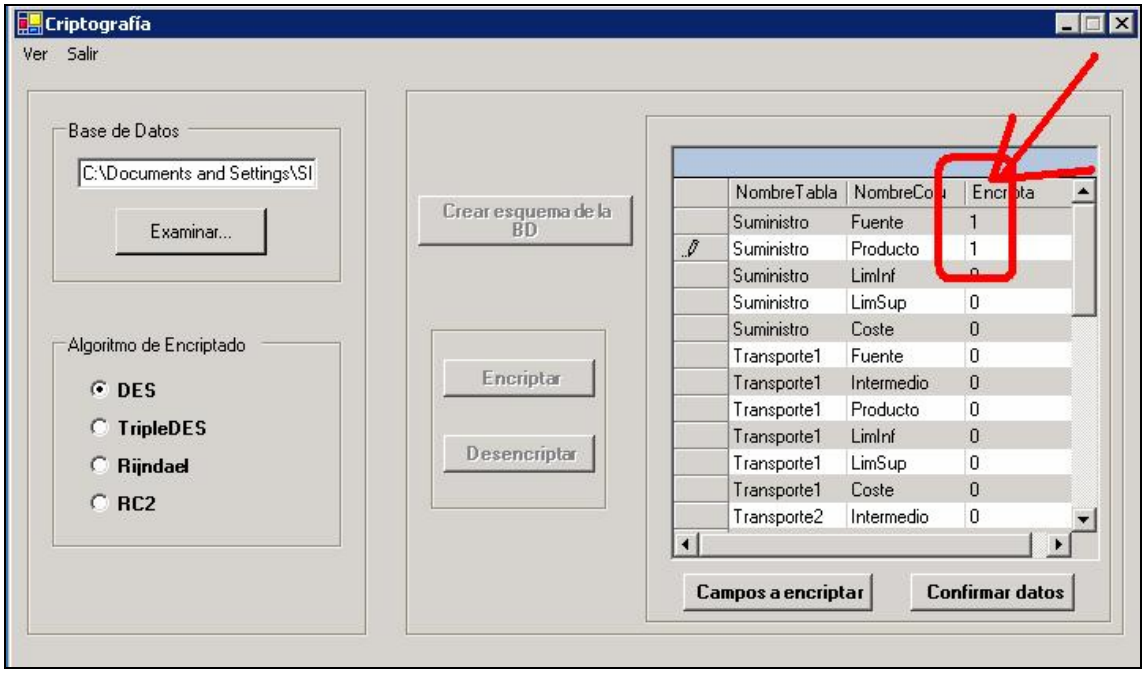
5. Pinchamos en campos a encriptar.



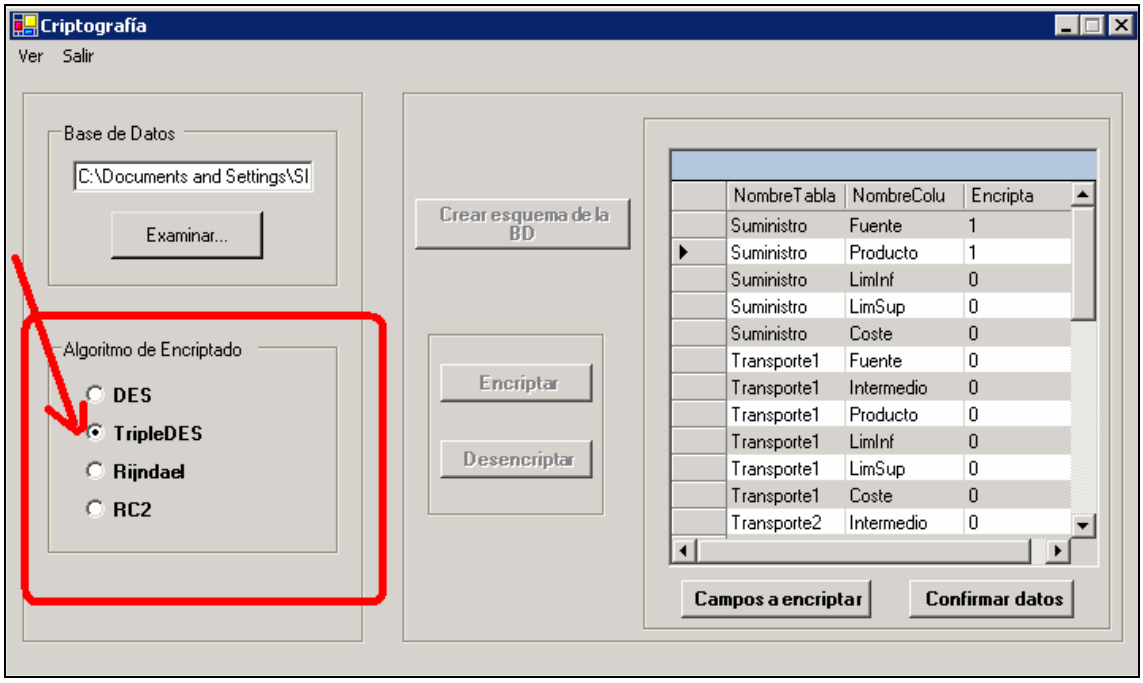
6. Observamos que en la ventana de arriba aparece una tabla donde las columnas corresponden a nombres de tablas, a nombres de columnas de esas tablas y finalmente a la opción de encriptar o no dicho campo. Por defecto, no se encripta ninguno.



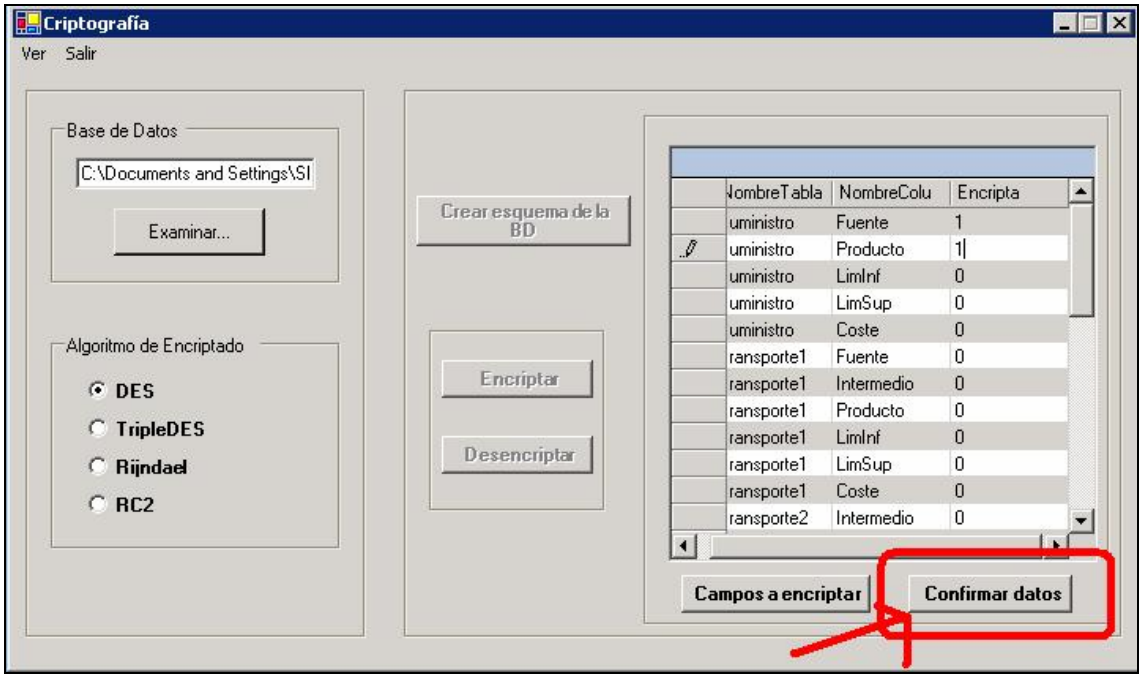
7. Si queremos encriptar un campo determinado (por ejemplo las dos primeras filas), no tenemos más que poner un “1” en la última columna (la que ahora hay “0”) de la fila que queramos codificar.



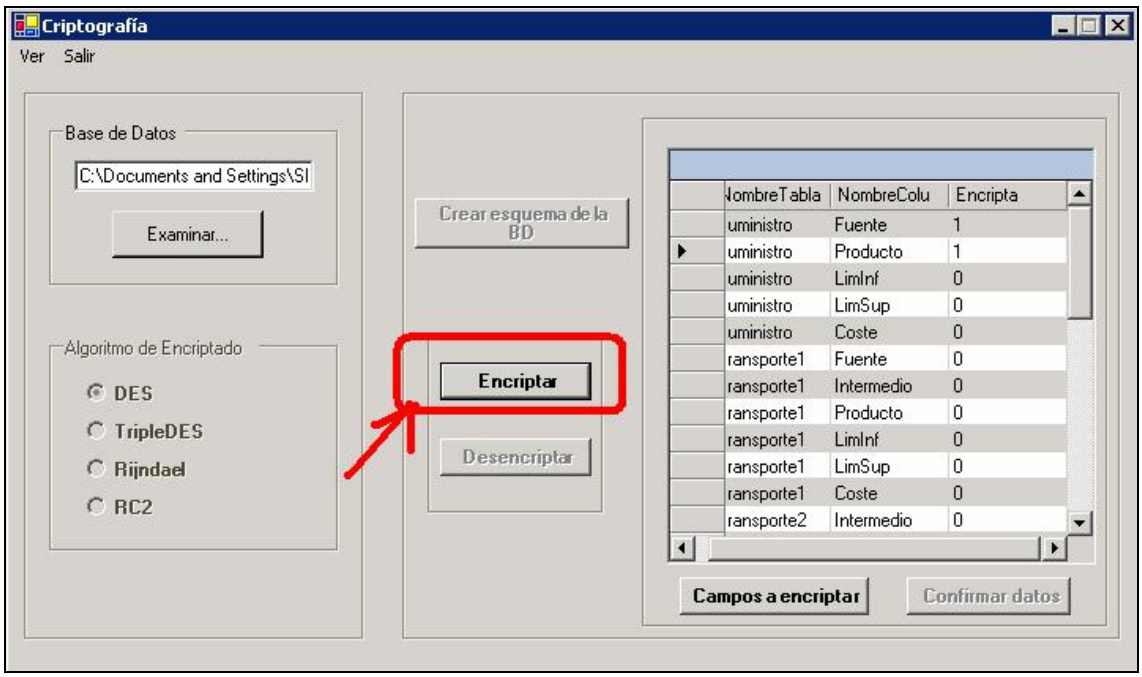
8. Por defecto, está puesto el algoritmo DES (a la izquierda de la aplicación). Si quisiéramos usar otro algoritmo, no tendríamos más que seleccionarlo de la lista que hay. En nuestro ejemplo, elegiremos el tripleDES.



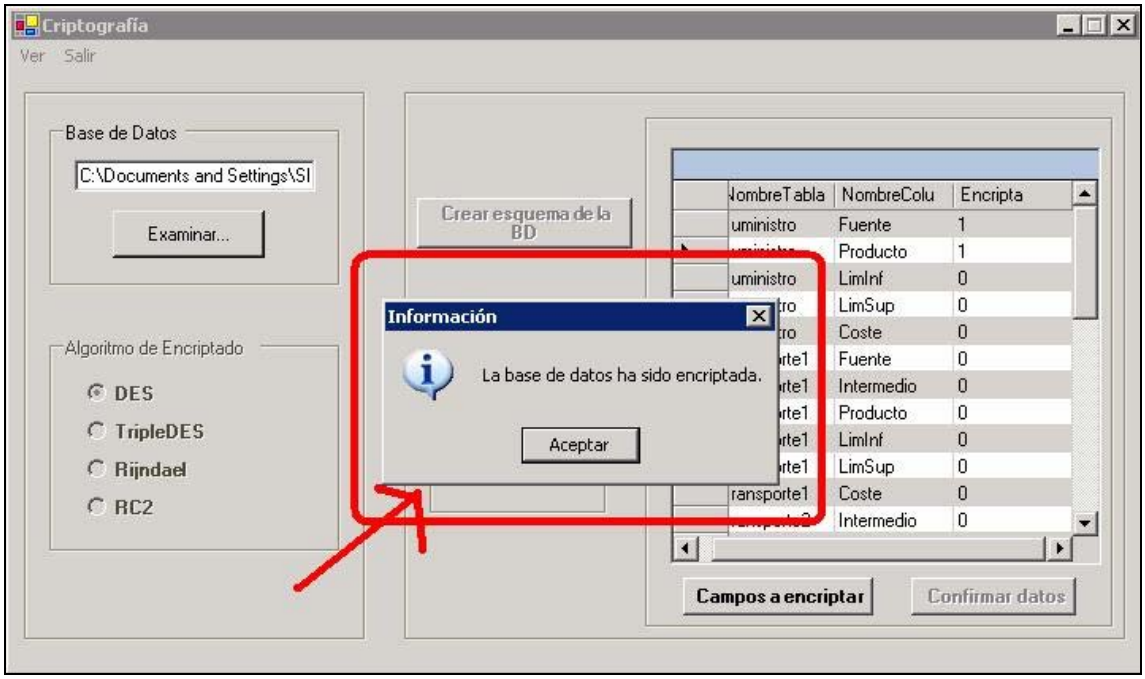
9. A continuación debemos confirmar los datos.



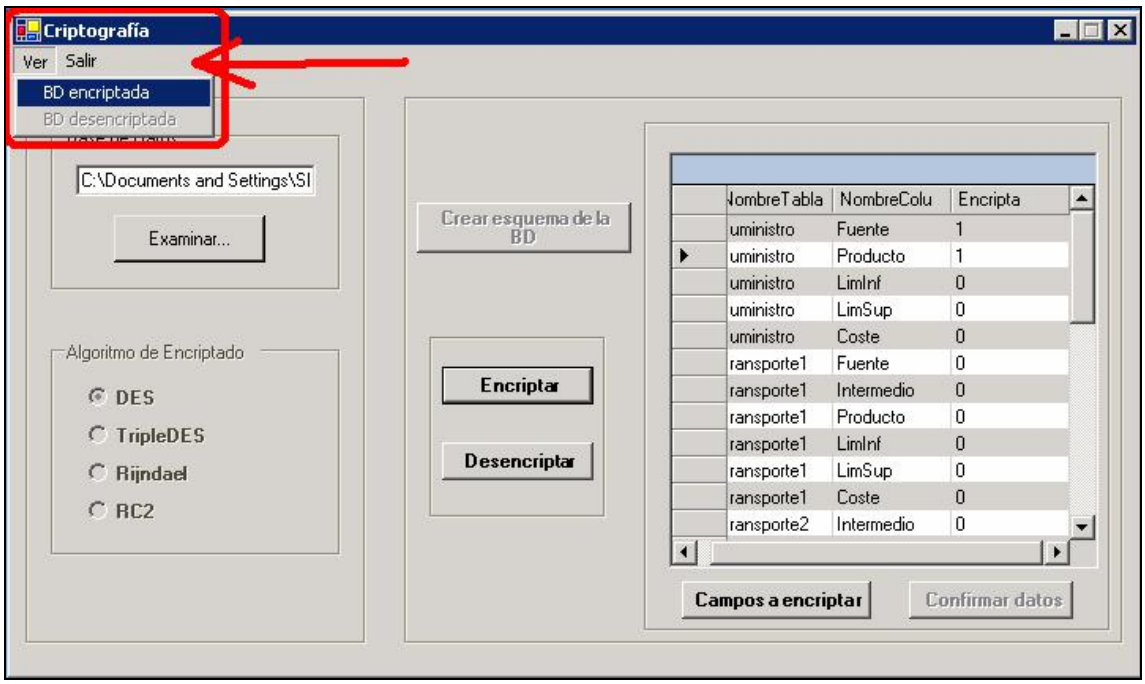
10. Se activará el botón “encriptar” (en el centro de la pantalla). Pinchamos sobre él.



11. La aplicación muestra la confirmación de que la base de datos ha sido encriptada.

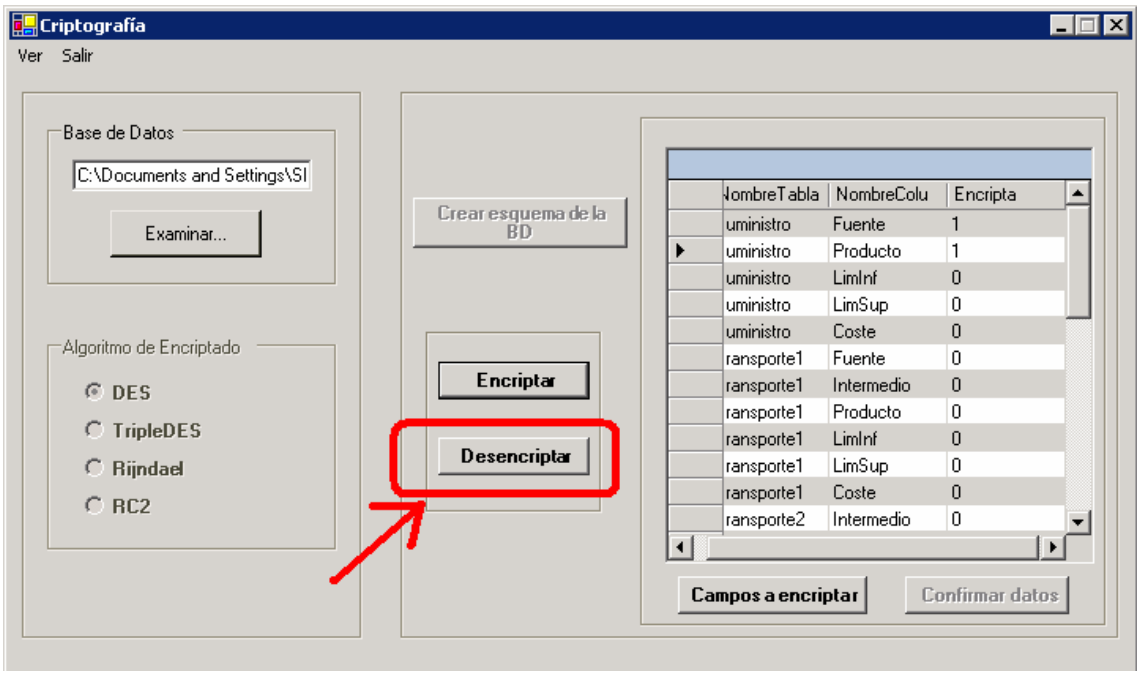


12. ¿Quieres ver cuál es el resultado de la codificación? Pincha en el menú "ver" en la opción "BD encriptada".

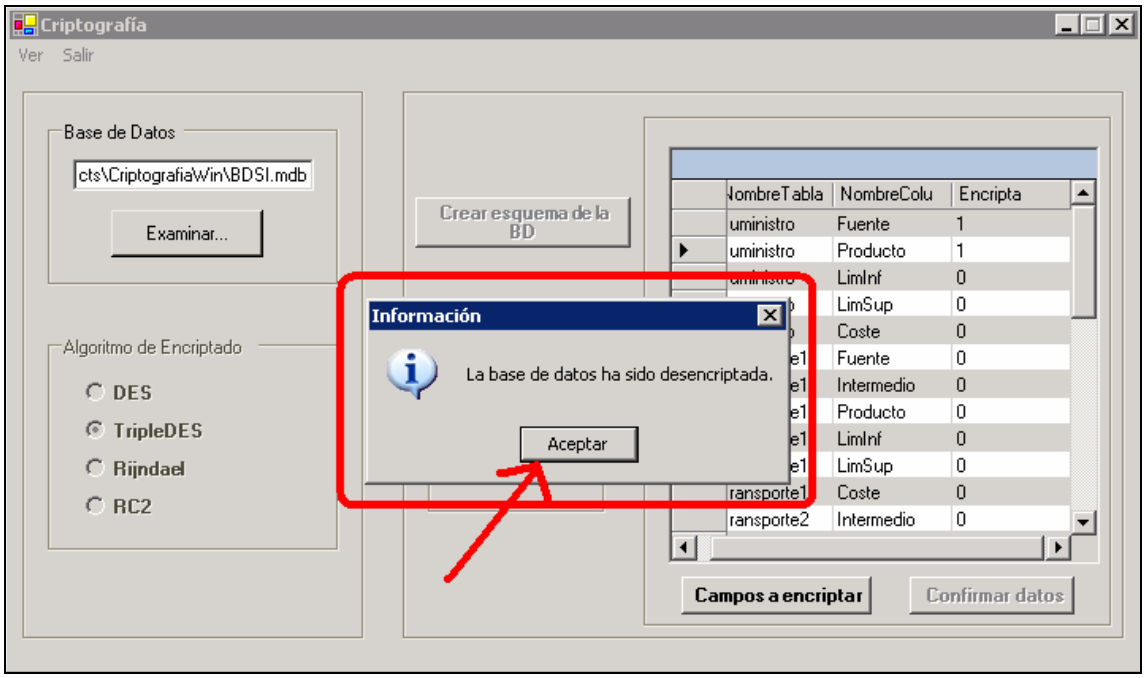


13..- Podrás visualizar en el navegador el código XML con los campos encriptados. En nuestro ejemplo:

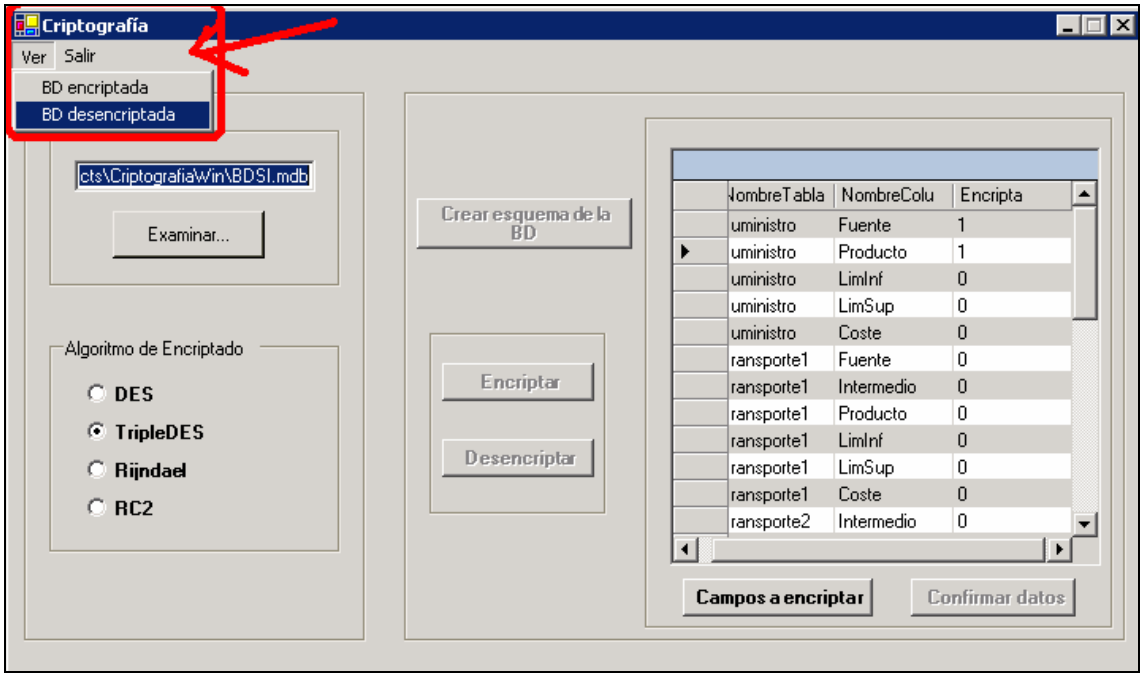
14. Si queremos desencriptar. Sencillamente pinchamos en “Desencriptar”



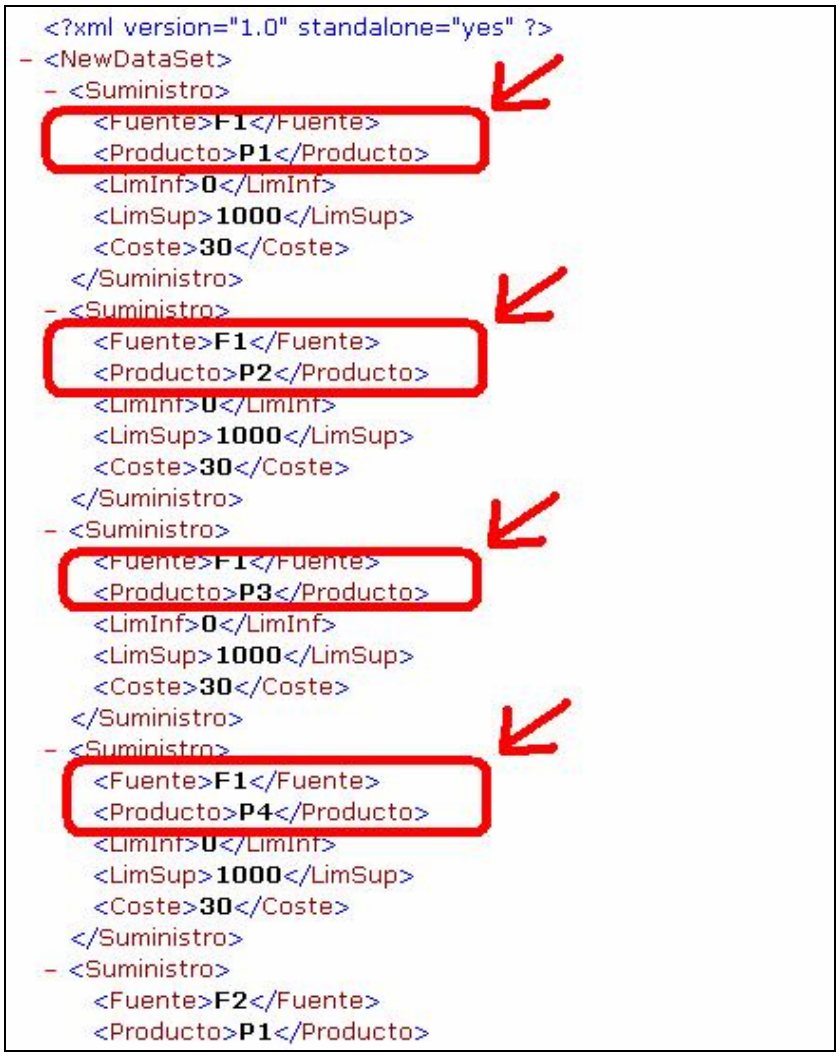
15. Aceptamos el aviso de que la base de datos ha sido desencriptada.



16. De nuevo, en el menú “ver” pinchamos esta vez en “BD Desencriptada”.



17. Y se nos muestra el resultado:


<?xml version="1.0" standalone="yes" ?>
<NewDataSet>
<Suministro>
<Fuente>F1</Fuente>
<Producto>P1</Producto>
<LimInf>0</LimInf>
<LimSup>1000</LimSup>
<Coste>30</Coste>
</Suministro>
<Suministro>
<Fuente>F1</Fuente>
<Producto>P2</Producto>
<LimInf>0</LimInf>
<LimSup>1000</LimSup>
<Coste>30</Coste>
</Suministro>
<Suministro>
<Fuente>F1</Fuente>
<Producto>P3</Producto>
<LimInf>0</LimInf>
<LimSup>1000</LimSup>
<Coste>30</Coste>
</Suministro>
<Suministro>
<Fuente>F1</Fuente>
<Producto>P4</Producto>
<LimInf>0</LimInf>
<LimSup>1000</LimSup>
<Coste>30</Coste>
</Suministro>
<Suministro>
<Fuente>F2</Fuente>
<Producto>P1</Producto>

10. ¿Quieres saber más?

10.1 Apéndice A. Seguridad: Criptografía.

10.1.1 Introducción.

La palabra criptografía proviene del griego kryptos, que significa esconder y gráphein, escribir, es decir, escritura escondida. La criptografía ha sido usada a través de los años para mandar mensajes confidenciales cuyo propósito es que sólo las personas autorizadas puedan entender el mensaje

Desde sus inicios la criptografía llegó a ser una herramienta muy usada en el ambiente militar, por ejemplo en la segunda gran guerra tuvo un papel determinante, una de las máquinas de cifrado que tuvo gran popularidad se llamó ENIGMA. Al terminar la guerra las agencias de seguridad de las grandes potencias invirtieron muchos recursos para su investigación. La criptografía como la conocemos hoy, surgió con la invención de la computadora.

La criptografía actual se inicia en la segunda mitad de la década de los años 70. No es hasta la invención del sistema conocido como DES (Data Encryption Standard) en 1976 que se da a conocer mas ampliamente, principalmente en el mundo industrial y comercial. Posteriormente con el sistema RSA (Rivest, Shamir, Adleman) en 1978, se abre el comienzo de la criptografía en un gran rango de aplicaciones: en transmisiones militares, en transacciones financieras, en comunicación de satélite, en redes de computadoras, en líneas telefónicas, en transmisiones de televisión etcétera.

La criptografía se divide en dos grandes ramas, la criptografía de clave privada o simétrica y la criptografía de clave pública o asimétrica, DES pertenece al primer grupo y RSA al segundo.

Alguien que quiere mandar información confidencial aplica técnicas criptográficas para poder "esconder" el mensaje.

Para poder entender un poco de la criptografía, es tiempo de plantear que tipo de problemas resuelve ésta. Los principales problemas de seguridad que resuelve la criptografía son: la privacidad, la integridad, la autenticación y el no rechazo.

La privacidad, se refiere a que la información sólo pueda ser leída por personas autorizadas.

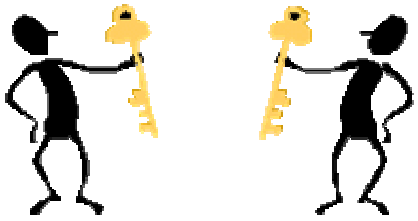
La integridad, se refiere a que la información no pueda ser alterada en el transcurso de ser enviada.

La autenticidad, se refiere a que se pueda confirmar que el mensaje recibido haya sido mandado por quien dice lo mando o que el mensaje recibido es el que se esperaba.

El no rechazo, se refiere a que no se pueda negar la autoría de un mensaje enviado.

10.1.2 Encriptación Simétrica.

La criptografía simétrica se refiere al conjunto de métodos que permiten tener comunicación segura entre las partes siempre y cuando anteriormente se hayan intercambiado la clave correspondiente que llamaremos clave simétrica. La simetría se refiere a que las partes tienen la misma llave tanto para cifrar como para descifrar.



Este tipo de criptografía se conoce también como criptografía de clave privada o criptografía de llave privada.

Existe una clasificación de este tipo de criptografía en tres familias, la criptografía simétrica de bloques (block cipher), la criptografía simétrica de lluvia (stream cipher) y la criptografía simétrica de resumen (hash functions). Aunque con ligeras modificaciones un sistema de criptografía simétrica de bloques puede modificarse para convertirse en alguna de las otras dos formas, sin embargo es importante verlas por separado dado que se usan en diferentes aplicaciones.

La criptografía simétrica ha sido la más usada en toda la historia, ésta ha podido ser implementada en diferente dispositivos, manuales, mecánicos, eléctricos, hasta los algoritmos actuales que son programables en cualquier computadora. La idea general es aplicar diferentes funciones al mensaje que se quiere cifrar de tal modo que solo conociendo una clave pueda aplicarse de forma inversa para poder así descifrar.
Aunque no existe un tipo de diseño estándar, quizá el más popular es el de Fiestel, que consiste esencialmente en aplicar un número finito de interacciones de cierta forma, que finalmente da como resultado el mensaje cifrado. Este es el caso del sistema criptográfico simétrico más conocido, DES.

DES

DES es un sistema criptográfico que toma como entrada un bloque de 64 bits del mensaje y este se somete a 16 interacciones, una clave de 56 bits, en la práctica el bloque de la clave tiene 64 bits, ya que a cada conjunto de 7 bits se le agrega un bit que puede ser usada como de paridad.

Dependiendo de la naturaleza de la aplicación DES tiene 4 modos de operación para poder implementarse: ECB (Electronic Codebook Mode) para mensajes cortos, de menos de 64 bits, CBC (Cipher Block Chaining Mode) para mensajes largos, CFB (Cipher Block Feedback) para cifrar bit por bit ó byte por byte y el OFB (Output Feedback Mode) el mismo uso pero evitando propagación de error.
En la actualidad no se ha podido romper el sistema DES desde la perspectiva de poder deducir la clave simétrica a partir de la información interceptada, sin embargo con un método a *fuerza bruta*, es decir probando alrededor de 256 posibles claves, se pudo romper DES en Enero de 1999. Lo anterior quiere decir que, es posible obtener la clave del sistema DES en un tiempo relativamente corto, por lo que lo hace inseguro para propósitos de alta seguridad. La opción que se ha tomado para poder suplantar a DES ha sido usar lo que se conoce como cifrado múltiple, es decir aplicar varias veces el mismo algoritmo para fortalecer la longitud de la clave, esto a tomado la forma de un nuevo sistema de cifrado que se conoce actualmente como triple-DES o TDES.

Además de DES y triple-DES, podemos destacar otros sistemas de cifrado como Rijndael y RC2.

Rijndael

Posee una estructura en capas formadas por funciones polinómicas reversibles (tienen inversa) y no lineales.
Utiliza una longitud de llave que está entre 128 y 256 bits (16 y 32 bytes). Para el vector de inicialización, la longitud será igual a 128bits (16 bytes).

RC2

Permite definir el tamaño del bloque a encriptar, el tamaño de la clave utilizada y el número de fases de encriptación.

A diferencia del anterior sistema, la longitud de llave estará entre 64 y 128 bits; y la del vector de inicialización será igual a 64 bits.

10.1.3 Encriptación Asimétrica.

La criptografía asimétrica es por definición aquella que utiliza dos claves diferentes para cada usuario, una para cifrar que se le llama clave pública y otra para descifrar que es la clave privada. El nacimiento de la criptografía asimétrica se dio al estar buscando un modo más práctico de intercambiar las llaves simétricas. Diffie y Hellman, proponen una forma para hacer esto, sin embargo no fue hasta que el popular método de Rivest Shamir y Adleman, RSA, publicado en 1978, cuando toma forma la criptografía asimétrica, su funcionamiento esta basado en la imposibilidad computacional de factorizar números enteros grandes.

Actualmente la Criptografía asimétrica es muy usada; sus dos principales aplicaciones son el intercambio de claves privadas y la firma digital. Una firma digital se puede definir como una cadena de caracteres que se agrega a un archivo digital que hace el mismo papel que la firma convencional que se escribe en un documento de papel ordinario. Los fundamentos de la criptografía asimétrica pertenecen a la teoría de números, algo de esto lo podemos ver en los textos A course in Number Theory and Cryptography y Algebraic Aspects of Cryptography de N. Koblitz, así como en Elementary Number Theory and Its Applications de K.H. Rosen.

En la actualidad la criptografía asimétrica o de clave pública se divide en tres familias según el problema matemático en el cual basan su seguridad. La primera familia es la que basa su seguridad en el Problema de Factorización Entera PFE, los sistemas que pertenecen a esta familia son, el sistema RSA, y el de Rabin Williams RW. La segunda familia es la que basa su seguridad en el Problema del Logaritmo Discreto PLD, a esta familia pertenece el sistema de Diffie Hellman DH de intercambio de claves y el sistema DSA de firma digital. La tercera familia es la que basa su seguridad en el Problema del Logaritmo Discreto Elíptico PLDE, en este caso hay varios esquemas tanto de intercambio de claves como de firma digital que existen como el DHE (Diffie Hellman Elíptico), DSAE, (Nyberg-Rueppel) NRE, (Menezes, Qu, Vanstone) MQV, etcétera.

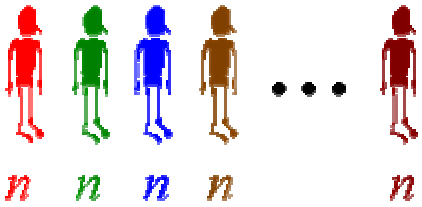
Sistema RSA.

En el caso de RSA, el problema matemático es el de la factorización de un número entero n grande (1024 bits, que equivale a un número de 308 dígitos), este número entero se sabe es producto de dos números primos p, q de la misma longitud, entonces la clave pública es el número n y la privada es p, q. El razonamiento del funcionamiento de RSA es el siguiente:

1.- A cada usuario se le asigna un número entero n, que funciona como su clave pública

2.- Solo el usuario respectivo conoce la factorización de n (o sea p,q), que mantiene en secreto y es la clave privada

3.- existe un directorio de claves públicas



4.- Si alguien quiere mandar un mensaje m a algún usuario entonces elige su clave pública n y con información adicional también pública puede mandar el mensaje cifrado c, que solo podrá descifrar el usuario correspondiente, el mensaje m convertido a número (codificación) se somete a la siguiente operación (donde e es constante y público)

$$c = m^e \bmod n$$

5.- Entonces el mensaje c puede viajar sin problema por cualquier canal inseguro cuando la información cifrada llega a su destino el receptor procede a descifrar el mensaje con la siguiente fórmula

6.- Se puede mostrar que estas fórmulas son inversas y por lo tanto dan el resultado deseado, (n,e) son la clave pública, la clave privada es la pareja (p,q) o equivalentemente el número d. La relación que existe entre d y e es que uno es el inverso multiplicativo del otro módulo l (n) donde l (n) es el mínimo común múltiplo de p-1 y q-1, o también puede usarse j (n)=(p-1)(q-1) esto significa que la clave privada o el la pareja p,q o es el número d.

$$m = c^d \bmod n$$

En términos muy generales es así como funciona el sistema RSA. Sin embargo en la realidad existen dos formas que son las más comunes, estas formas dependen de la aplicación y se llaman el esquema de firma y el esquema de cifrado.

10.1.4 Estándar de seguridad ws.security.

Los desarrolladores y los arquitectos de software son conscientes de que, en su intento por reemplazar a los antecesores distribuidos, la implementación de servicios Web exige la creación de arquitecturas cada vez más complejas. Puesto que las especificaciones diseñadas para admitir estas arquitecturas experimentan una evolución constante, los juegos de herramientas que se adaptan a dichas especificaciones resultan esenciales para garantizar la interoperabilidad y la productividad del desarrollador. Web Services Enhancements 2.0 para Microsoft .NET (WSE 2.0) es un complemento de Visual Studio .NET y .NET Framework que permite a los desarrolladores satisfacer los complejos requisitos empresariales. WSE se utiliza para aplicar credenciales de nivel de mensaje, cifrado y firmas digitales a los mensajes SOAP independientes del protocolo de transporte. Mediante la creación de estos principios básicos de seguridad, se pueden establecer relaciones de confianza a través de varios extremos entre los que no exista una confianza directa.

WSE v.2.0 representa un gran avance en lo que respecta a la conformidad y la interoperabilidad en el marco de la iniciativa de los servicios Web. Basado en WS-Security, WSE 2.0 mejora la implementación de la seguridad, la confianza y las conversaciones seguras en la arquitectura de servicios Web

10.2 Apéndice B. Programación Evolutiva: Optimizar.

10.2.1 Introducción.

En la naturaleza todos los seres vivos se enfrentan a problemas que deben resolver con éxito, como conseguir más luz del sol, o cazar una mosca. La Programación Evolutiva interpreta la naturaleza como una inmensa máquina de resolver problemas y trata de encontrar el origen de dicha potencialidad para utilizarla en nuestros programas.

Los Algoritmos Genéticos son una de las más conocidas y originales técnicas de resolución de problemas dentro este paradigma de programación.

Un Algoritmo Evolutivo es una técnica de resolución de problemas inspirada en la evolución de los seres vivos.

En un Algoritmo Evolutivo se define una estructura de datos que admita todas las posibles soluciones a un problema.

Cada uno de los posibles conjuntos de datos admitidos por esa estructura será una solución al problema. Unas soluciones serán mejores, otras peores.

Solucionar el problema consistirá en encontrar la solución óptima, y por tanto, los Algoritmos Evolutivos son en realidad un método de búsqueda.

Pero un método de búsqueda muy especial, en el que las soluciones al problema son capaces de reproducirse entre sí, combinando sus características y generando nuevas soluciones.

En cada ciclo se seleccionan las soluciones que más se acercan al objetivo buscado, eliminando el resto de soluciones. Las soluciones seleccionadas se reproducirán entre sí, permitiendo de vez en cuando alguna mutación o modificación al azar durante la reproducción.

Dentro de los paradigmas de aprendizaje automático, los algoritmos genéticos se presentan como uno de los más prometedores en el camino hacia la inteligencia artificial. Actualmente han demostrado su validez en múltiples áreas de aplicación, y en cualquier caso, poseen un gran interés por la evidente analogía que convierte al programador en un creador de "vida".

10.2.2 Pasos que realiza un algoritmo genético.

1. Se genera un conjunto de 1-N soluciones válidas al problema. Valores típicos de N son desde 1 hasta 200. Cada una de estas entidades representa una solución distinta a un mismo problema. Esta(s) entidades se pueden generar al azar. También se pueden generar a partir de soluciones ya conocidas del problema, que se pretendan mejorar, o mediante posibles "trozos de soluciones" (más conocidos como bloques constructores), es decir, con lo que creemos que pueden ser elementos componentes de la solución final aunque no sepamos cómo combinarlos.
2. Se evalúan las soluciones existentes, y se decide, en función de esta evaluación, dos cosas. Por una parte, cuáles soluciones van a sobrevivir y cuáles no; y por otra, cuáles se van a reproducir y cuáles no. En el caso de reproducirse, se especifica la potencia reproductora de la solución, de forma que es posible decidir que unas soluciones se reproduzcan más que otras.
3. Tal como se ha establecido en el paso anterior, se eliminan ciertas soluciones y se mantienen otras, y se efectúa la reproducción o recombinación de genes (normalmente por parejas) de las entidades existentes. Por ejemplo, se realizan cruzamientos de patrones a partir de cierto punto elegido al azar, de forma que los nuevos patrones posean un segmento de cada uno de los progenitores.
4. Se efectúan mutaciones (cambios al azar en los genes) de los nuevos patrones, según una tasa determinada. Algunos estudios aconsejan realizar mutaciones también sobre los padres.
5. Se continúa en el paso 2 hasta que se cumpla el criterio de parada, que puede ser por ejemplo, que el peso o *fitness* de la mejor entidad supere cierto valor.

10.2.3 ¿Cuándo se pueden aplicar los Algoritmos Genéticos?

En primer lugar, en el problema a resolver, la meta ha de poder ser observada en grados cualitativamente comparables. Por otra parte, las principales dificultades a la hora de implementar un algoritmo genético son:

- Definir una estructura de datos que pueda contener patrones que representen, la solución óptima buscada (desconocida) y todas las posibles alternativas de aproximaciones a la solución.
- Definir un tipo de patrón tal que si un patrón es seleccionado positivamente, que esto no sea debido a la interacción de los distintos segmentos del patrón, sino que existan segmentos que por sí solos provocan una selección positiva.
- Definir una función de evaluación que seleccione los mejores individuos.

Las condiciones que debe cumplir un problema para ser abordable por algoritmos evolutivos son:

- El espacio de búsqueda deber ser acotado.
- Debe existir un procedimiento relativamente rápido que asigne un grado de utilidad a cada solución propuesta, de forma que este grado de utilidad asignado corresponda, o bien directamente con la calidad de la solución en cuanto al problema a resolver, o bien con un valor de calidad relativo al resto de la población que permita obtener en el futuro mejores soluciones.
- Debe existir un método de codificación de soluciones que admita la posibilidad de que los cruces combinen las características positivas de ambos progenitores. Este método debe permitir también aplicar algún mecanismo de mutación que sea capaz de conseguir tanto soluciones muy dispares respecto de la solución sin mutar, como muy parecidas a ésta.

10.2.4 Tipos de Algoritmos Evolutivos.

La Computación Evolutiva (CE) es un término relativamente nuevo que intenta agrupar un batiburrillo de paradigmas muy relacionados cuyas competencias no están aún muy definidas. Hasta hace poco era común hablar de Algoritmo Genético (AG) en general, en vez de identificar diferentes tipos de CE, ya que el resto de los algoritmos se pueden interpretar como variaciones o mejoras de los algoritmos genéticos, más conocidos. En un Algoritmo Genético los elementos de las cadenas (genes) son típicamente bits, y en ciertos casos, valores numéricos o strings. Por su parte, en las Estrategias Evolutivas los elementos de las cadenas son valores reales que actúan como reglas que definen cómo va a actuar el individuo ante cierta situación, y en la Programación Genética son instrucciones en un lenguaje de programación. Simulated Annealing se puede considerar una simplificación de los AG cuyo origen está en los procedimientos físicos de solidificación controlada. Los Clasificadores Genéticos solucionan problemas de reconocimiento de patrones mediante un entrenamiento basado en ejemplos, almacenando en las cadenas información que relacione los datos de entrada y la salida deseada. Finalmente, por razones conceptuales a mí me gusta considerar la Vida Artificial como un superconjunto de todas estas técnicas, aunque en la práctica constituyen disciplinas estancas

10.2.5 Opciones de un Algoritmo Evolutivo.

En la siguiente tabla se resumen algunas opciones de un algoritmo evolutivo. Podemos idear muchas más. En general se trata de distintas formas de producir, o bien una convergencia más rápida hacia una solución, o bien una exploración más a fondo del espacio de búsqueda. Ambas cosas son

deseables y contradictorias, por lo que se ha de llegar a un compromiso. Este compromiso es lo que antes he llamado cooperación. Por supuesto que todo esto no son más que metáforas. Se requiere de una fuerza conservadora (explotación-egoísmo), que beneficie a los mejores agentes, es decir, a los que mejor resuelvan nuestro problema. Esto es evidente, pero no basta. También es necesaria una fuerza innovadora (exploración-altruismo), que permita la existencia de agentes muy distintos, aún cuando su peso sea menor. Así se puede obtener la variedad suficiente para evitar una población estancada en un máximo local, y permitir la resolución de problemas cambiantes o con varios máximos. Esto ocurre de forma espontánea en la naturaleza por ser algo inherente a cada entidad, que puede ser seleccionado, pero resulta más cómodo programarlo de forma externa, es decir, haremos que nuestro programa seleccione para la reproducción "los agentes buenos", pero también "unos cuantos de los malos" para mantener la variedad.

Opciones Generales
- Número de entidades.
- Número de elementos (genes, reglas) por cada agente.

Método de Evaluación: Asignar un peso
- Desordenar las entidades antes de evaluarlas
- Diferentes formas de modificación de los pesos después de la evaluación. Por ejemplo, el peso de una entidad se puede calcular independientemente de las demás entidades, o se puede modificar posteriormente este valor, disminuyendo el peso si existe otra entidad muy parecida, analizando para ello un cierto subconjunto de la población vecina.

Método de Selección: ¿Quién muere? ¿Quién se reproduce?
- Con o sin reemplazamiento
- Método de la ruleta
- Método de los torneos
- Seleccionar el n% mejor y el m% peor

Método de Reproducción: Generar y mutar nuevos hijos
- Los padres pueden tomarse por parejas o en grupos más numerosos, elegidos al azar o en orden de pesos.
- En el caso de detectar que los progenitores son muy parecidos, se puede realizar una acción especial, como incrementar la probabilidad de mutación.
- Las entidades pueden comunicar a otras su conocimiento, ya sea a toda o a una parte de la población, directamente o a través de una pizarra, (una especie de tablón de anuncios).
- Método de recombinación de genes: se puede tomar genes de uno u otro progenitor al azar, en un cierto orden, con uno, dos o más puntos de corte, etc.
- Tasa de mutación variable.
- Fijar una tasa de mutación diferente para cada individuo o incluso para cada gen.
- Hacer que sea más probable que se produzca una mutación en un gen si en su vecino ya se ha producido.
- Sustituir por mutaciones genes sin utilidad, como reglas incorrectas o

repetidas.
- Tipos de mutaciones

10.2.6 El problema de la variedad.

Un algoritmo genético trata de explorar las regiones más prometedoras de un enorme espacio de posibilidades. Al encontrar una zona con pesos altos, ésta se explora más a fondo. Pero hay que evitar que el algoritmo se estanque en una determinada zona, produciendo multitud de cadenas muy parecidas. De igual forma, se ha de evitar lo contrario, que el algoritmo distribuya tan uniformemente el espacio de búsqueda que casi se dejen de explorar las mejores zonas. Se trata de encontrar un equilibrio, la mejor de las opciones en cada caso.

10.2.7 Soluciones al problema de la variedad.

Se trata de conseguir un equilibrio entre exploración y explotación, o la velocidad óptima de convergencia del algoritmo, para que siga las direcciones más prometedoras pero sin olvidar otras posibilidades que a la larga pueden producir mejores resultados; llegar al optimo rápidamente sin pero quedarse estancado en un máximo local.

El algoritmo genético básico o canónico, en el que siempre se seleccionan los mejores individuos, tiende por lo general a la homogeneización de la población. Se trata por tanto de aumentar la variedad seleccionando algunos de los individuos que no son los mejores.

El aumento de variedad se consigue con las mutaciones, pero esto no suele bastar, y no es aconsejable excederse en las mutaciones. Una posible solución sería seleccionar un gran número de agentes, para evitar que un pequeño grupo repita sus características en una gran población, pero esto no es suficiente a largo plazo.

Lo que se debe hacer es seleccionar, además de la población con mayor peso, una fracción de la de menor peso. Si sólo seleccionáramos unos pocos de los mejores, los muy buenos, a no ser que el problema sea muy simple, el algoritmo probablemente nunca funcione como esperamos, ya que se quedará estancado en un máximo local. La especialización es peligrosa. La analogía con la vida es inevitable.

10.2.8 El problema de la reproducción.

Las características positivas de los individuos deben poder transmitirse a la descendencia.

En cada ciclo de un algoritmo genético se selecciona un subconjunto de las soluciones o individuos existentes, eliminando el resto. Los individuos seleccionados se reproducirán entre sí, generando nuevas soluciones.

Se trata de elegir las mejores soluciones, para que al reproducirse, generen otras nuevas que combinen los aspectos positivos de cada progenitor. El problema es encontrar un mecanismo de combinación de información de dos (o más) individuos, de manera que el (o los) individuos resultantes posean, al menos en la mayoría de los casos, tantas o más cualidades positivas que sus progenitores, y por tanto se encuentren tanto o más cercanos al objetivo que ellos.

En el caso de no existir reproducción, sólo mutaciones, también debemos encontrar un tipo de mutaciones que en un número suficiente de casos permitan mantener e incluso aumentar la bondad de la entidad mutada.

Sin embargo, en muchos problemas es muy difícil dar con una estructura de datos y un modo de reproducción que se comporte de esta forma. Por el contrario, con frecuencia ocurre que en la combinación de soluciones no sólo no se mantienen las cualidades positivas de los progenitores, sino que además se generan con frecuencia soluciones no válidas, cuya aproximación al objetivo buscado es nula.

10.2.9 Solución al problema e la reproducción.

El problema de la reproducción aparece cuando la estructura de datos elegida no es la adecuada. En la mayoría de los casos, las estructuras de datos más sencillas e intuitivas no contienen información acerca de porqué esa solución ha sido seleccionada, y por tanto no es posible transmitir a la descendencia características positivas de los individuos progenitores ya que éstas no están representadas en ningún lugar, y cualquier método de reproducción elegido no producirá los efectos deseados.

Ésta parece una dificultad insalvable, pero no lo es. Una vez más, la vida natural nos da la pista: en realidad, los genes, más que determinar directamente las características que definen un nuevo individuo (color de los ojos, etc.), definen un conjunto de reglas de desarrollo que aplicadas y combinadas provocarán dichas características. Para ello, la dotación genética, que existe en todas las células de cada ser vivo, controla la realización de ciertas reacciones químicas, de manera que los genes son los que determinan, pero no directamente, la constitución y comportamiento de dicho individuo.

La solución al problema de la reproducción es encontrar una estructura de datos capaz de contener la información crítica necesaria para alcanzar el objetivo propuesto. Aquí entramos en el tema de los métodos de representación del conocimiento ampliamente estudiado en los sistemas expertos. En muchos casos, es suficiente un sistema de reglas.

10.2.10 El problema de la selección.

En cada ciclo de un algoritmo genético se selecciona un subconjunto de las soluciones o individuos existentes, eliminando el resto. Los individuos seleccionados se reproducirán entre sí, generando nuevas soluciones.
La función que decide qué entidades serán seleccionadas, la llamada función de evaluación (FE), puede ser muy difícil o imposible de conseguir. ¿Cómo podemos saber cuáles son las mejores soluciones? ¿Cuáles son las que más se acercan al objetivo buscado? ¿Cuáles producirán una convergencia hacia la resolución del problema?

El problema se agrava cuando el algoritmo genético está dirigido por un objetivo que no posee naturaleza más o menos continua. Cuando el objetivo puede ser logrado en diversos grados, el algoritmo puede esforzarse en conseguir que el objetivo se cumpla en el máximo grado posible, detectando pequeñas variaciones. Pero si el estado buscado fuera todo-nada, es decir, si ocurre que: una de dos, o se consigue el objetivo completamente o no se consigue en absoluto, no es posible que la evolución produzca gradualmente entidades cada vez más cercanas al objetivo

10.2.11 Soluciones al problema de la selección.

La función de evaluación más corriente consiste en asignar a cada solución un peso o valor en función del grado en que se acerca al objetivo. Una vez hecho esto, se ordenan todas las soluciones según este criterio. Finalmente, se selecciona un número determinado de soluciones, comenzando por las primeras de la lista.
En el problema del viajante, que debe recorrer varias ciudades hasta volver al punto de partida incurriendo en el mínimo coste, la FE calculará la suma de las distancias entre las ciudades recorridas en el orden especificado y devolverá la inversa de este valor de forma que los recorridos cortos posean un peso mayor que los largos.

Sin embargo, como se trata de seleccionar unas soluciones y eliminar otras, no es necesario conocer *cuantitativamente* (con un número) el grado en el que cada solución se acerca a la solución buscada. Basta con conocer *cualitativamente* qué soluciones se aproximan más que otras.

De esta forma, podemos agrupar las soluciones por criterios de proximidad física o azar y hacer que "jueguen", "peleen" o demuestren de alguna forma cuáles son las mejor preparadas en función del objetivo buscado.

10.2.12 Tipos de aplicaciones.

Cualquier problema puede verse como un problema de algunos de estos tipos:

- Búsqueda

- Clasificación
- Predicción
- Optimización (Parametrización, Configuración, Maximización, Minimización)

10.2.13 Algoritmo Evolutivo como un método de optimización.

Interpretar un algoritmo evolutivo como un método de optimización es lo más intuitivo. La función de evaluación devuelve valores altos para las soluciones buenas, con lo que cada vez tendremos mejores soluciones, y en eso consiste precisamente optimizar.

10.3 Apéndice C. XML: Conceptos básicos.

10.3.1 Qué es XML.

XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores.

XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Así pues, el XML juega un papel importantísimo en este mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología que permitirá compartir la información de una manera segura, fiable, fácil. Además, XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

XML no es un lenguaje, sino varios lenguajes, no es una sintaxis, sino varias y no es una manera totalmente nueva de trabajar, sino una manera más refinada que permitirá que todas las anteriores se puedan comunicar entre si sin problemas, ya que los datos cobran sentido.

10.3.2 Historia del XML.

El XML proviene de un lenguaje que inventó IBM allá por los años 70. El lenguaje de IBM se llama GML (General Markup Language) y surgió por la necesidad que tenían en la empresa de almacenar grandes cantidades de información de temas diversos.

Imaginar por un momento la cantidad de documentación que generaría IBM sobre todas las áreas en las que trabajaba e investigaba, y la cantidad de información que habrá generado hasta hoy. Así pues, necesitaban una manera de guardar la información y los expertos de IBM se inventaron GML, un lenguaje con el que poder clasificarlo todo y escribir cualquier documento para que se pueda luego procesar adecuadamente.

Este lenguaje gustó mucho a la gente de ISO, una entidad que se encarga de normalizar cuantas cosas podáis imaginar para los procesos del mundo actual, de modo que allá por el 86 trabajaron para normalizar el lenguaje, creando el SGML, que no era más que el GML pero estándar (Standar en inglés).

SGML es un lenguaje muy trabajado, capaz de adaptarse a un gran abanico de problemas y a partir de él se han creado los siguientes sistemas para almacenar información.

Por el año 89, para el ámbito de la red Internet, un usuario que había conocido el lenguaje de etiquetas (Markup) y los hiperenlaces creo un nuevo lenguaje llamado HTML, que fue utilizado para un nuevo servicio de Internet, la Web. Este lenguaje fue adoptado rápidamente por la comunidad y varias organizaciones comerciales crearon sus propios visores de HTML y riñeron entre ellos para hacer el visor más avanzado, inventándose etiquetas como su propia voluntad les decía. Desde el 96 hasta hoy una entidad llamada W3C ha tratado de poner orden en el HTML y establecer sus reglas y etiquetas para que sea un estándar. Sin embargo el HTML creció de una manera descontrolada y no cumplió todos los problemas que planteaba la sociedad global de Internet.

El mismo W3C en el 98 empezó y continúa, en el desarrollo de XML (Extended Markup Language). En este lenguaje se ha pensado mucho más y muchas personas con grandes conocimientos en la materia están trabajando todavía en su gestación. Pretendían solucionar las carencias del HTML en lo que se respecta al tratamiento de la información. Problemas del HTML como:

El contenido se mezcla con los estilos que se le quieren aplicar.

No permite compartir información con todos los dispositivos, como pueden ser ordenadores o teléfonos móviles.
La presentación en pantalla depende del visor que se utilice.

Imagínese, una persona que conoce el HTML y lo difícil que puede llegar a ser entender su código, que tuviese que procesarlo para extraer datos que necesite en otras aplicaciones. Sería muy difícil saber dónde está realmente la información que busca, siempre mezclada entre etiquetas <FONT>, <TABLE>, <TD>, etc. Esto es una mala gestión de la información y el XML la soluciona.

10.3.3 Sintaxis del XML.

Dicen que el XML es un 10% del SGML y de verdad lo es, porque en realidad las normas que tiene son muy simples. Se escribe en un documento de texto ASCII, igual que el HTML y en la cabecera del documento se tiene que poner el texto

```
<?xml version="1.0"?>
```

En el resto del documento se deben escribir etiquetas como las de HTML, las etiquetas que nosotros queramos, por eso el lenguaje se llama XML, lenguaje de etiquetas extendido. Las etiquetas se escriben anidas, unas dentro de otras.

```
<ETIQ1>...<ETIQ2>...</ETIQ2>...</ETIQ1>
```

Cualquier etiqueta puede tener atributos. Le podemos poner los atributos que queramos.

<ETIQ atributo1="valor1" atributo2="valor2"...>

Los comentarios de XML se escriben igual que los de HTML.

<!-- Comentario -->

Y esto es todo lo que es el lenguaje XML en sí, aunque tenemos que tener en cuenta que el XML tiene muchos otros lenguajes y tecnologías trabajando alrededor de él. Sin embargo, no cabe duda que la sintaxis XML es realmente reducida y sencilla.

Para definir qué etiquetas y atributos debemos utilizar al escribir en XML tenemos que fijarnos en la manera de guardar la información de una forma estructurada y ordenada. Por ejemplo, si deseamos guardar la información relacionada con una película en un documento XML podríamos utilizar un esquema con las siguientes etiquetas.

```
<?xml version="1.0"?>
<PELICULA nombre="El Padrino" año=1985>
<PERSONAL>
<DIRECTOR nombre="Georgie Lucar"/>
<INTERPRETE nombre="Marlon Brando" interpreta-a="Don Corleone"/>
<INTERPRETE nombre="Al Pacino" interpreta-a="Michael Corleone"/>
</PERSONAL>
<ARGUMENTO descripción="Pelicula de mafias sicilianas en Estados Unidos"/>
</PELICULA>
```

10.3.4 Contenidos: DTD o XML Schema

Un documento XML puede contener muchos tipos de información. Es decir, pueden haber muchos lenguajes escritos en XML para cualquier colectivo de usuarios

Como vemos, se pueden crear infinitos lenguajes a partir del XML. Para especificar cada uno de los usos de XML, o lo que es lo mismo, para especificar cada uno de los sublenguajes que podemos crear a partir de XML, se utilizan unos lenguajes propios.

Son unos lenguajes que sirven para definir otros lenguajes, es decir, son metalenguajes. Los definen especificando qué etiquetas podemos o debemos encontrarnos en los documentos HTML, en qué orden, dentro de qué otras, además de especificar los atributos que pueden o deben tener cada una de las etiquetas.

Hay dos metalenguajes con los que definir los lenguajes que podemos obtener a partir de XML, el DTD y el XML Schema.

El DTD, Definition Type Document, tiene una sintaxis especial, distinta de la de XML, que es sencilla, aunque un poco rara si nunca hemos visto un documento similar.

Para evitar el DTD, que tiene una sintaxis muy especial, se intentó encontrar una manera de escribir en XML la definición de otro lenguaje XML. Se definió entonces el lenguaje XML Schema y funciona bien, aunque puede llegar a ser un poco más complicado que especificarlo en DTD. Simplemente nos ahorramos de aprender un nuevo lenguaje con su sintaxis particular.

10.3.5 Diseño: CSS o XSL.

Para cada documento XML que se desee presentar en pantalla formateado de la manera que deseemos se tiene que escribir una hoja de estilos o similar.

También tenemos dos posibles lenguajes con los que formatear los textos de un documento XML para poder verlo por pantalla. La primera posibilidad es el CSS y la segunda, el XSL, bastante más avanzada.

10.3.6 Programación: SAX o DOM.

Si queremos realizar acciones con nuestros datos escritos en XML tenemos también mucho camino ya implementado. El W3C ha especificado dos mecanismos para acceder a documentos XML y trabajar con ellos. Se tratan simplemente de unas normas que indican a los desarrolladores la manera de acceder a los documentos. Estas normas incluyen una jerarquía de objetos que tienen unos métodos y atributos con los que tendremos que trabajar y que nos simplificarán las tareas relativas al recorrido y acceso a las partes del documento.

Estos dos mecanismos se denominan **SAX y DOM**. SAX se utiliza para hacer un recorrido secuencial de los elementos del documento XLM y DOM implica la creación de un árbol en memoria que contiene el documento XML, y con él en memoria podemos hacer cualquier tipo de recorrido y acciones con los elementos que queramos.

11. Glosario.

NET
Es el relanzamiento con un enfoque distinto al actual de la informática distribuida tipo Internet o Intranet, dejando a un lado herramientas de acceso que se han tenido como estándares, tipo DCOM (Distributed Component Object Model) o CORBA (Commot Object Request Broker Architecture) y sustituyéndolos por otros con mayor facilidad de utilización, como XML, HTTP o SOAP.

ASP
Active Server Pages o páginas activas de servidor. Probablemente la sustitución de las HTML tal como ahora se conocen por una diferencia básica, estas últimas son estáticas en tanto que por medio de programación no se modifiquen, mientras que las ASP son analizadas por el servidor modificando el código HTML según las necesidades.

ASP (Aplicaciones)
Application Service Providers o servicio de provisión de aplicaciones. Es un sistema de utilización de programas de muy reciente implantación, en el cual lo que el cliente paga es la utilización, no la compra, de tal forma, que se establece normalmente una conexión con un servidor ajeno, que es el propietario de los programas.

Autentificación
Verificación de la identidad de una persona o de un proceso en orden de acceder a un recurso o poder realizar una determinada actividad. También se aplica a la verificación de identidad de origen de un mensaje.

Base de Datos
Conjunto de ficheros dedicados a guardar información relacionada entre sí, con referencias entre ellos de manera que se complementen con el principio de no duplicidad de datos. Dependiendo de cómo se vinculen dan lugar a B.D. jerárquicas, relacionales, etc. Un caso especial de éstas son las documentales, que, como su nombre indica, están diseñadas para almacenar volúmenes grandes de documentos, lo que genera una problemática distinta por los sistemas de búsqueda.

Certificación
Procedimiento por el cual una entidad o un particular garantiza que un determinado dato (una firma electrónica o una clave pública) pertenece realmente a quien se supone.

Certificado
Acreditación emitida por una entidad o un particular debidamente autorizada garantizando que un determinado dato (una firma electrónica o una clave pública) pertenece realmente a quien se supone.

Clave (key)

Código de signos convenidos para la transmisión de mensajes secretos o privados.

Cliente - Servidor
Se le suele llamar así a la arquitectura a dos capas, es decir, una capa servidor, u ordenador que contendrá los datos y los programas gestores asociados, y capas clientes, u ordenadores que se dirigirán al anterior para obtener la información.

COM
Component Object Model Estas tecnologías, originarias de Microsoft, sirven para el diseño de componentes, en realidad de Objetos programables, y la base de OLE tanto en sus variantes OLE DB (DataBase) como OLE DS (Directory Services) y ActiveX

Criptografía
Ver Encriptación o Sistemas de Encriptación o Cifrado.

Criptología
Campo de la Criptografía que tiene por objeto el descifrado de criptogramas cuando se ignora la clave.

Datagrama
Entidad de datos autocontenida e independiente que transporta información suficiente en orden de ser encaminada desde su ordenador de origen a su ordenador de destino sin tener que depender de que se haya producido anteriormente tráfico alguno entre ambos y la red de transporte.

DCOM
Proviene de COM Distribuido. Diseñado para el acceso a componentes COM a través de redes.

DTD
Definición de tipo de documento. Especifica el conjunto de elementos que puede incluir un fichero XML, ya sea propio del mismo, o externo, y puede acompañar, generalmente como cabecera, al XML, siendo en este caso un DTD interno, o asociar ambos ficheros, estando en el caso de un DTD externo. Además, pueden ser privados, que se encuentran en una ubicación específica, generalmente de un particular, o de carácter mucho más genérico, utilizados por comunidades de usuarios, que se denominan públicos

EFS
Encription File System. Sistema de encriptación de ficheros, introducido en Windows 2000 como forma de protección del contenido de los archivos. Es una forma muy simple a la que se accede con solo aceptar esa propiedad en la configuración del equipo, opciones de seguridad, y muy delicado por la facilidad de perder los datos por reinstalaciones o problemas similares.

Encriptación
Encriptar es hacer ilegible un escrito por medio de aplicar al texto un algoritmo.

Enrutador
Router. Se denomina así al dispositivo capaz de dirigir la información, dividida en paquetes, por el camino más idóneo, examinando la dirección y el destino y utilizando mapas de red.

Firma electrónica
Versión, a nivel de comunicaciones, de la firma digital tradicional.

Heurística
Es una técnica que se basa en la experiencia conseguida después de realizar intentos repetidos, normalmente por medio de algoritmos concretos. Se van realizando pruebas, aún sin conocer exactamente todos los datos a tener en cuenta, hasta conseguir una solución al problema. Es uno de los sistemas de funcionamiento de los antivirus.

Hojas de estilo
Generalmente referidas a la programación Web, una hoja de estilo es la que alberga el formato que se va a utilizar: el tipo de letra, los márgenes, tamaños de los componentes, en definitiva, como va a ser cada componente. Entre las formas más comunes que se utilizan en la actualidad son las CSS (Cascading Style Sheets), hojas de estilo en cascada, que es la normalizada para documentos HTML

HTML
Hypertext Markup Language. Al redactar un escrito en un procesador de textos normal, lo que se ve en pantalla no es lo que realmente se graba en el disco. Si comprobásemos el trasfondo de lo escrito, aparecerían caracteres por todas partes ilegibles, pero que sirven al procesador para crear la apariencia de lo que se ve. Cuando hablamos de páginas Web ocurre mas o menos igual. Tras la apariencia de ésta misma página se esconde la realidad de lo que hay escrito, su color, fondo, tipo de letra, etc. Esto se consigue a través del lenguaje HTML, u otros.

HTTP
Es el protocolo o las reglas de funcionamiento de los servidores WWW, que son los encargados de mantener este tipo de páginas.

HTTPS
Creado por Netscape Communications Corporation para designar documentos que llegan desde un servidor WWW seguro. Esta seguridad es dada por el protocolo SSL (Secure Sockets Layer) basado en la tecnología de encryptación y autentificación desarrollada por la RSA Data Security Inc.

IA o Inteligencia Artificial
Es una de las partes de la informática. En ella se pretende (o pretendía) un comportamiento del ordenador similar al que pudiese elaborar la mente humana. Aún siendo un proceso complejo, su base teórica es mas o menos simple, se basa en premisas y reglas que devuelven unos hechos mediante

algoritmos, siendo el programa capaz de memorizar o aprender de los resultados. Se considera su creador a Alan M. Turing.

Interfaz
Este término se utiliza con distintas acepciones,. Principalmente es un lugar físico común entre dos dispositivos informáticos y que permite la conexión entre ellos. No obstante se habla de interfaz gráfica, de usuario, etc. y no tiene una relación con lo explicado.

Interfaz de usuario
Es la manera de funcionar el ordenador de cara al usuario, o mejor, la relación de ambos, es decir, cómo responde a los sucesos o acciones.

Internet
Se le pueden dar definiciones puntuales, como "red de redes" que aunque no nos sirve para entender nada, está bien, pues en definitiva no es más que redes de ordenadores interconectadas. Como va más allá de esta realidad, vamos a comentarlo con algo más de detenimiento. Históricamente tiene su origen en ARPANET, proyecto militar estadounidense, en 1.969. A partir de los años 80 se extiende al mundo científico y en la actualidad es universal. Uno de los puntos importantes es el uso de un protocolo propio, TCP-IP, donde IP son las siglas de Protocolo de Internet, y también se denominan así a las direcciones de los servidores, son unas series de números que los identifican y que, para comodidad del usuario, se traducen a nombres, como cualquiera que se puede ver en una página de enlaces o ésta misma donde nos encontramos. A estos "apodos" se les denomina direcciones DNS. Otro de los aspectos que han influido en la universalización de Internet es el sistema WWW o Web, tanto es así que prácticamente se identifica uno con el otro a nivel popular.

ISO
International Standardization Organization. Es el organismo de estandarización internacional de nivel más alto, con sede en Ginebra. Es responsable de las OSI.

ODBC
Open Data Base Conectivity. Ha sido la base de Windows en sistemas abiertos, es decir que permiten una conectividad entre distintos lenguajes de programación con distintas bases de datos.

RC2
Algoritmo criptográfico de clave privada (simétrico).

RSA
Algoritmo criptográfico de clave pública y amplia utilización el cual está patentado por los autores que le dan nombre.

RSS
RSS es un formato de archivos basado en XML. Podemos decir que es nuevo y hasta la fecha su utilización se ha dado preferentemente en publicación de noticias, al cual se puede acceder a través de programas lectores de noticias

sin necesidad de abrir su navegador de Internet El formato RSS tiene diferentes versiones, de las cuales las más comunes son 0.91, 1.0, y últimamente 2.0. La compatibilidad no parece asegurada.

Servicios
Se le suele denominar así al conjunto de posibilidades que tiene un ordenador, generalmente el servidor, y que puede distribuir entre los otros ordenadores que así lo reclamen.
Servidor
Se denomina así al ordenador que se encarga de suministrar lo necesario a una red, dependiendo de cual sea la finalidad de ésta.

Servidor Web
Computadora dedicada a gestionar el uso de la red por otras computadoras llamadas clientes la cual contiene archivos y recursos que pueden ser accedidos desde otras computadoras o terminales.

SGBD
Sistema Gestor de Bases de Datos. Conjunto de programas que hacen posible la creación y mantenimiento de una base de datos. En estos momentos la tendencia es a las Bases de Datos relacionales basadas en lenguaje de interrogación SQL, y aunque se utiliza uno de sus estándares cada fabricante introduce sus modificaciones. Pero no tiene por qué ser así, un SGBD o DBMS puede ser interrogado desde muy distintos lenguajes de programación e incluso por combinaciones entre estos y SQL, o tener los suyos propios. En cualquier caso todos funcionan a través de lo que denominan "motores" de datos. Algunos de ellos, aún siendo pequeños, tienen capacidad gracias a su *Administrador de datos* de poder generar archivos, editarlos, imprimirlos, etc. sin necesidad de programación.

SGML
Generalized Markup Language. Es la norma de edición electrónica más importante. Desarrollada por IBM en 1.969 parte de dos bases: el concepto de tipo de documento y la introducción de marcas en el mismo según su contenido y estructura. De él parte XML.

Sistemas de Encriptación o Cifrado
La criptografía es un área gigante, porque su origen es muy antiguo y sistemas de cifrado hay muchos, pero existe una división básica: Criptografía de Clave Privada, legible tan sólo por el destinatario que conoce la forma de descifrarlo, a diferencia de la Criptografía de Clave Pública, que puede serlo por distintos destinatarios, ya que en realidad son dos claves y una persona (organismo) certificadora. Cualquier sistema de encriptación o cifrado es un sistema matemático. Cuantos más bits pueda utilizar, es un efecto potencial, mucho más difícil puede ser la ecuación o el sistema que integre, por lo tanto más seguro, pero también más lento.

Sistemas de Información
Se debe considerar un sistema de computación e información como el conjunto de componentes físicos (hardware), lógicos (software), de comunicación (bien

redes de cualquier tipo o Internet) y medios humanos (lo que ahora llaman orgware), todo ello unido permite el tratamiento de la información.

Sistemas Expertos
Está dentro del apartado Inteligencia Artificial. Se podría decir que planteado un problema, éste es el encargado de analizar y sacar los resultados por medio de lo que se conoce como "Motor de Inferencia".

SOAP
Simple Object Access Protocol. Creado por Microsoft para programar servicios en la Web que se basan en XML, con independencia del lenguaje de programación y del Sistema Operativo.

TCP-IP
Transmision Control Protocol-Internet Protocol. Protocolo en el que se basa Internet y que en realidad consiste en dos. El TCP, especializado en fragmentar y recomponer paquetes, e IP para direccionarlos hasta su destino.

Transmisiones
Referido puramente a las informáticas, envío de información entre dos computadoras, normalmente vía módems. Los datos se envían troceados, lo que se llaman tramas. Suelen utilizarse dos tipos, las asíncronas y las síncronas. En el primer caso, para que los módems "se entiendan" es necesario incluir en cada trama unos bits de control, en el segundo no son necesarios pues cada trama se identifica de las demás por un período de tiempo, para ello se utiliza el reloj del ordenador, y saben que cada cierta cantidad de pulsos han de emitir y recibir. Además se incluyen distintos sistemas que sirven para corrección de errores, desde la paridad, que es lo más simple, hasta otros más complejos.

Transparencia
Se dice de los procesos sobre los que el usuario no tiene conocimiento. Se producen sin su intervención en el proceso en sí mismo.

UML
Se le denomina así al Lenguaje Unificado de Modelado, basados en los primeros métodos de Programación Orientada a Objetos (POO) y está pensado para realizar análisis completos para desarrollo de aplicaciones de unas dimensiones amplias.

UML (modelado)
Unified Modeling Languaje. Es la unificación de metodologías de análisis y diseño destinadas a Objetos. Conlleva una "notación" o sintaxis en el modelado, ya sea a través de los diagramas o de cualquier forma, y un "metamodelo" que es un diagrama de clases representativo del método orientado a objetos, es decir, la representación de las clases y los objetos junto con sus relaciones.

URL

Uniform Resource Locator. Se conoce por este nombre a las direcciones dentro de Internet, normalmente, aunque no necesariamente, refiriéndonos a páginas Web. En este caso se distinguen por iniciarse con http:// No obstante es una simplificación para el usuario el referenciarlas de esta forma, en realidad son secuencias de números que se dirigen de forma inequívoca a una dirección. Esto se conoce como DNS.

Web
Por éste término se suele conocer a WWW (World Wide Web), creado por el Centro Europeo de Investigación Nuclear como un sistema de intercambio de información y que Internet ha estandarizado. Supone un medio cómodo y elegante, basado en multimedia e hipertexto, para publicar información en la red. Inicial y básicamente se compone del protocolo http y del lenguaje html. Un ejemplo de páginas de éste tipo, es la que tienes delante en estos momentos.

WML
Wireless Markuo Language. Es un desarrollo del lenguaje HTML dirigido al protocolo WAP, o en realidad y dadas las características del medio para el que se usa (telefonía movil y PDA's), de una derivación de este lenguaje de programación Web, que se denomina XML.

XML
Extensive Markup Language. Parecido a HTML pero más moderno y flexible. Se creó en 1.998 por el World Wide Wed Consortium (conocido por W3C) como sustituto del anterior, pensando principalmente en los negocios en la Red. Es muy simple de utilizar y con unas características de hiperenlaces muy potentes gracias a las especificaciones XLL (Extended Linking Language). Lo curioso de XML es que ha sido creado para desaparecer, la W3C busca fusionarle con HTML para dar lugar a XHTML.

XQL
Extensible Query Language. Existía un lenguaje de interrogación que era XSL (figuradamente como un SQL cara a documentos Webs), lo que hace de nuevo es añadirle tres booleanos para las búsquedas, and, or y not.

XSLT
Extensible Stylesheet Language - Transformations. Es una especificación pensada para transformar el formato XML en cualquier otro, aunque el evidente debe de ser el HTML Para la transformación se incluye el texto dentro de unas marcas, y , reservadas, que serán utilizadas por las órdenes o elementos XSL.

12. Bibliografía.

12.1 Básica.

- "Profesional C# 2ª Edición". Simon Robinson, K. Scott Allen, ... Ed: Wrox.Press 2002
- "Profesional UMLwith Visual Studio .NET", Filev, Loton,... Ed: Wrox Press 2002
- "Profesional Web Service Security". Galbrait, Hankinson, Hiotis, ... Ed Wrox Press 2002
- http://wwwdi.ujaen.es/~mlucena
- http://www.programacion.com/tutorial/csharp/
- http://www.programacionfacil.com/csharpcgi/indice.htm
- http://geneura.ugr.es/~jmerelo/ws/
- http://www.guajara.com/wiki/es/wikipedia/c/co/computacion_evolutiva.html
- http://www.fdi.ucm.es/datos/Docu_Docente2.asp
- http://www.kriptopolis.com
- http://msdn.microsoft.com/library/en-us/
- http://www.dotnet.com

12.2 Complementaria.

- http://www.elguille.info/NET/indice.asp
- http://www.clikear.com/manuales/csharp/index.asp
- http://www.microsoft.com/spanish/msdn/comunidad/uni.net/
- http://www.dat.etsit.upm.es/~abarbero/curso/xml/xmltutorial.html
- http://www.htmlweb.net/seguridad/
- http://www.edicom.es/edi_internet.html
- http://www.microsoft.com/spanish/msdn/articulos/archivo/141103/voices/wssecdrill.asp
- http://www.redcientifica.com/gaia/ce/ceno_c.htm#quees
- http://www.programacion.com/html/xml/
- http://www.desarrolloweb.com/manuales/18/
- http://glosario.panamacom.com/?id_c=8
- http://www.phptr.com/
- http://www.informit.com/articles/article.asp?p=102212&seqNum=4
- http://www.awprofessional.com/articles/article.asp?p=29901&seqNum=1